

Contributing to Open Source Projects

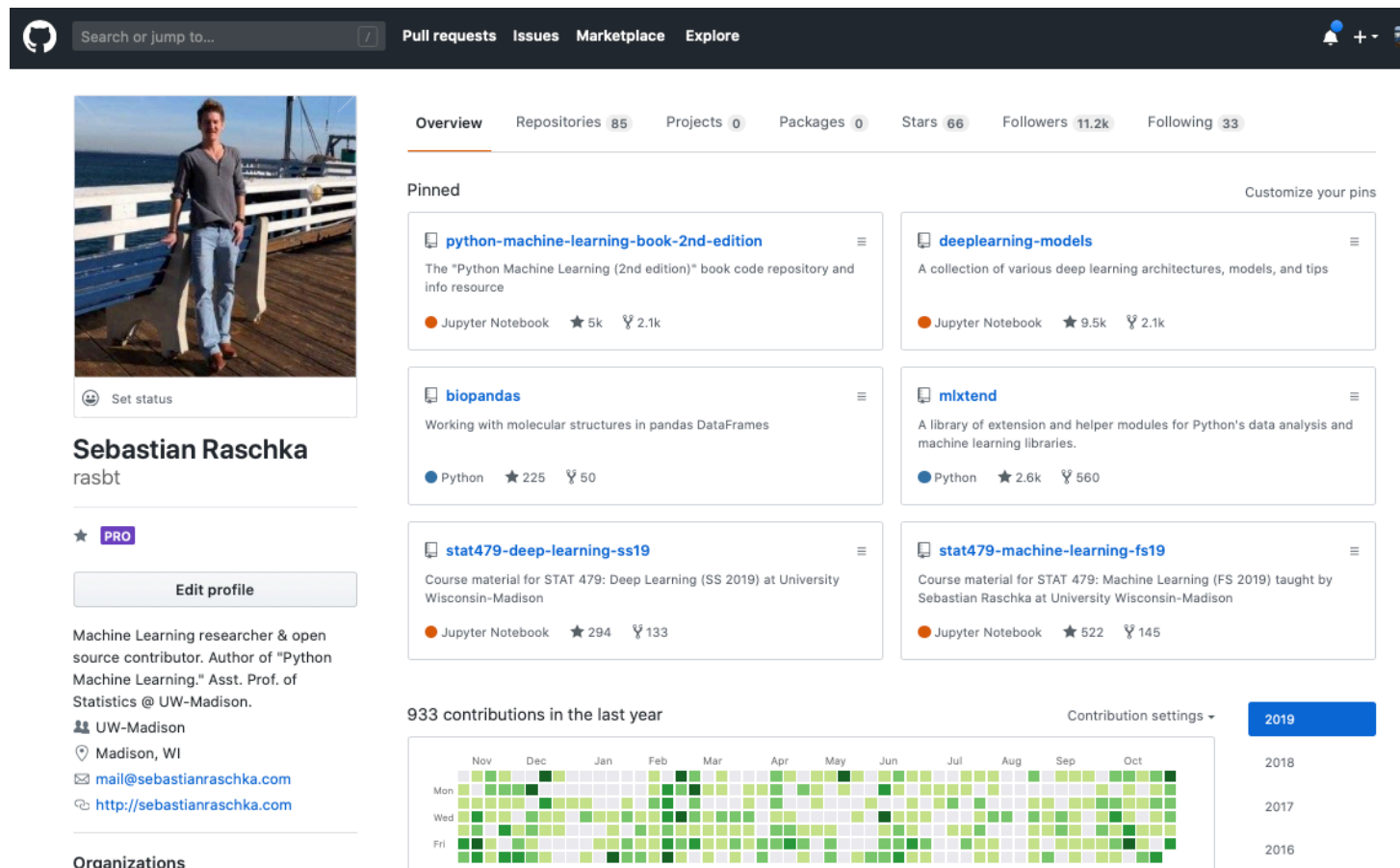
(like scikit-learn, mlxtend, etc.)

First Step: Familiarize yourself with Git & GitHub

Git vs. GitHub

Git is a software tool

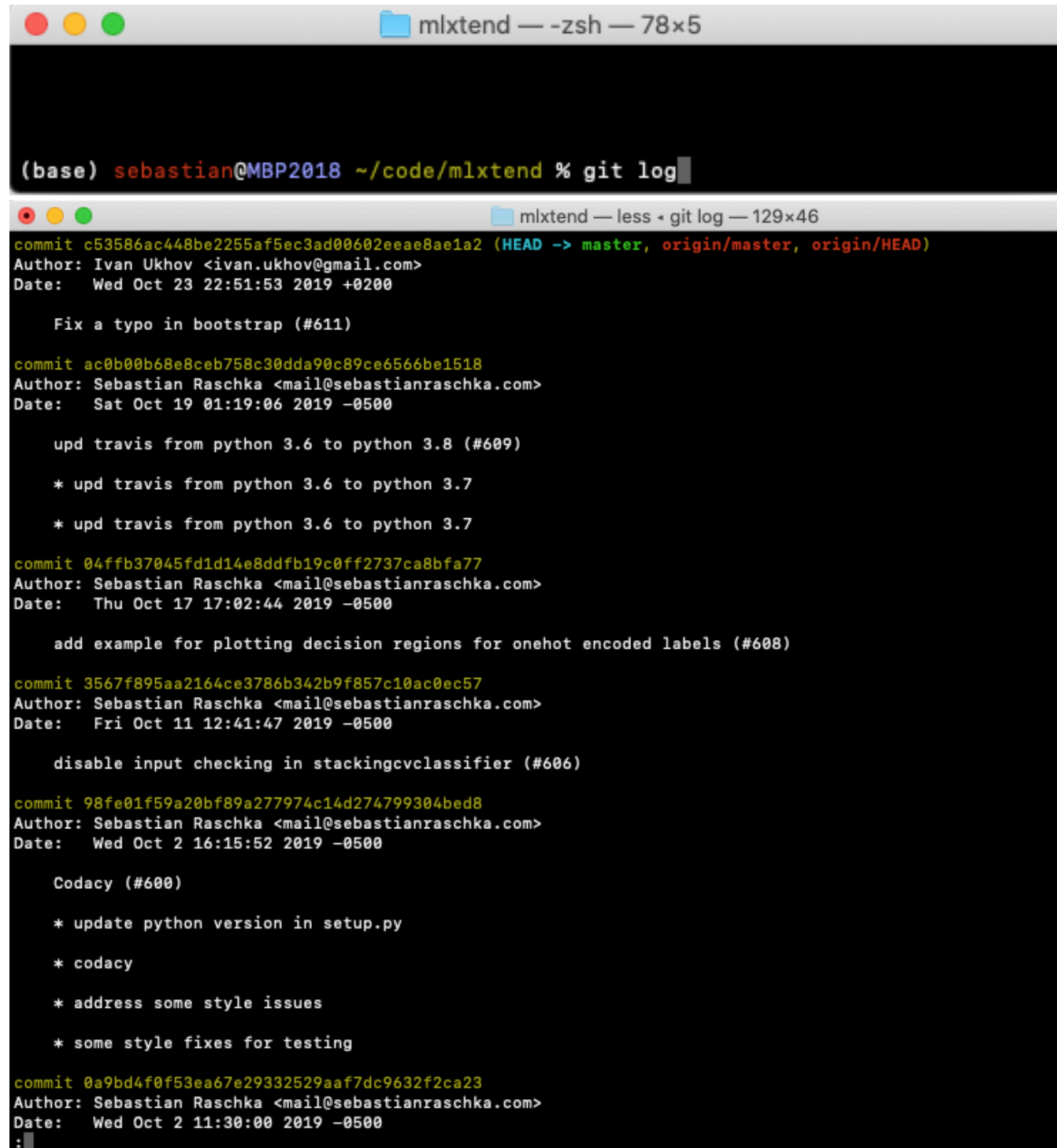
```
sebastian — zsh — 86x17
Last login: Fri Nov  1 11:41:22 on ttys000
(base) sebastian@MBP2018 ~ % git --version
git version 2.21.0 (Apple Git-122)
(base) sebastian@MBP2018 ~ %
```



GitHub is a (web) platform for hosting "Git" projects

Git

- A version control system for software development
- Let's you keep track of changes in your code base



```
(base) sebastian@MBP2018 ~/code/mlxtend % git log

commit c53586ac448be2255af5ec3ad00602eeae8ae1a2 (HEAD -> master, origin/master, origin/HEAD)
Author: Ivan Ukhov <ivan.ukhov@gmail.com>
Date:   Wed Oct 23 22:51:53 2019 +0200

    Fix a typo in bootstrap (#611)

commit ac0b00b68e8ceb758c30dda90c89ce6566be1518
Author: Sebastian Raschka <mail@sebastianraschka.com>
Date:   Sat Oct 19 01:19:06 2019 -0500

    upd travis from python 3.6 to python 3.8 (#609)

    * upd travis from python 3.6 to python 3.7

    * upd travis from python 3.6 to python 3.7

commit 04ffb37045fd1d14e8ddfb19c0ff2737ca8bfa77
Author: Sebastian Raschka <mail@sebastianraschka.com>
Date:   Thu Oct 17 17:02:44 2019 -0500

    add example for plotting decision regions for onehot encoded labels (#608)

commit 3567f895aa2164ce3786b342b9f857c10ac0ec57
Author: Sebastian Raschka <mail@sebastianraschka.com>
Date:   Fri Oct 11 12:41:47 2019 -0500

    disable input checking in stackingcvclassifier (#606)

commit 98fe01f59a20bf89a277974c14d274799304bed8
Author: Sebastian Raschka <mail@sebastianraschka.com>
Date:   Wed Oct 2 16:15:52 2019 -0500

    Codacy (#600)

    * update python version in setup.py

    * codacy

    * address some style issues


    * some style fixes for testing

commit 0a9bd4f0f53ea67e29332529aaf7dc9632f2ca23
Author: Sebastian Raschka <mail@sebastianraschka.com>
Date:   Wed Oct 2 11:30:00 2019 -0500

    :
```

Git

- Moreover, it's great for collaborative code development
- I.e., it's great for managing software development that involves multiple people
- A great learning resource is the free "Git" book at <https://git-scm.com/book/en/v2>



The screenshot shows the Git website's 'Book' section. At the top, the Git logo is followed by the tagline '--local-branching-on-the-cheap'. A search bar is located in the top right corner. On the left sidebar, there are links for 'About', 'Documentation' (highlighted in red), 'Reference', 'Book' (highlighted in red), 'Videos', and 'External Links'. Below these are sections for 'Downloads' and 'Community'. The main content area is titled 'Book' and contains a paragraph about the Pro Git book by Scott Chacon and Ben Straub, published by Apress. It mentions the Creative Commons Attribution Non Commercial Share Alike 3.0 license and provides a link to Amazon.com. To the right of the text is a book cover for 'Pro Git' 2nd Edition (2014). Below the book cover is a 'Download Ebook' section with icons for PDF, EPUB, and MOBI formats. The book is organized into two main parts: '1. Getting Started' and '2. Git Basics', each with a list of sub-topics.

git --local-branching-on-the-cheap

Search entire site...

About

Documentation

Reference

Book

Videos

External Links

Downloads

Community

This book is available in [English](#).

Full translation available in

[български език](#),

[Español](#),

[Français](#),

[Ελληνικά](#),

[日本語](#),

[한국어](#),

[Nederlands](#),

[Русский](#),

[Slovenščina](#),

[Tagalog](#),


[Українська](#)

Partial translations available in

[Čeština](#),




Book

The entire Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, is available here. All content is licensed under the [Creative Commons Attribution Non Commercial Share Alike 3.0 license](#). Print versions of the book are available on [Amazon.com](#).



2nd Edition (2014)

Download Ebook



1. Getting Started

- 1.1 [About Version Control](#)
- 1.2 [A Short History of Git](#)
- 1.3 [What is Git?](#)
- 1.4 [The Command Line](#)
- 1.5 [Installing Git](#)
- 1.6 [First-Time Git Setup](#)
- 1.7 [Getting Help](#)
- 1.8 [Summary](#)

2. Git Basics

- 2.1 [Getting a Git Repository](#)
- 2.2 [Recording Changes to the Repository](#)
- 2.3 [Viewing the Commit History](#)
- 2.4 [Undoing Things](#)
- 2.5 [Working with Remotes](#)
- 2.6 [Tagging](#)
- 2.7 [Git Aliases](#)
- 2.8 [Summary](#)

Additional Git Resources

1) What is Git | What is GitHub | Git Tutorial | GitHub Tutorial | Devops Tutorial | Edureka
<https://www.youtube.com/watch?v=xuB1Id2Wxak>

2) Understanding branches in Git
<https://blog.thoughttram.io/git/rebase-book/2015/02/10/understanding-branches-in-git.html>

3) .1 Git Branching - What a Branch Is
<https://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

4) Forking Workflow
<https://www.atlassian.com/git/tutorials/comparing-workflows/forking-workflow>

Basic Workflow for (most) Open Source Projects

Step 0: Read the Contributor Guidelines

<http://rasbt.github.io/mlxtend/CONTRIBUTING/>



How to Contribute

[Quick Contributor Checklist](#)

Tips for Contributors

[Getting Started - Creating a New Issue and Forking the Repository](#)

[Syncing an Existing Fork](#)

[*The Main Workflow - Making Changes in a New Topic Branch](#)

Notes for Developers

[Building the documentation](#)

[Uploading a new version to PyPI](#)

How to Contribute

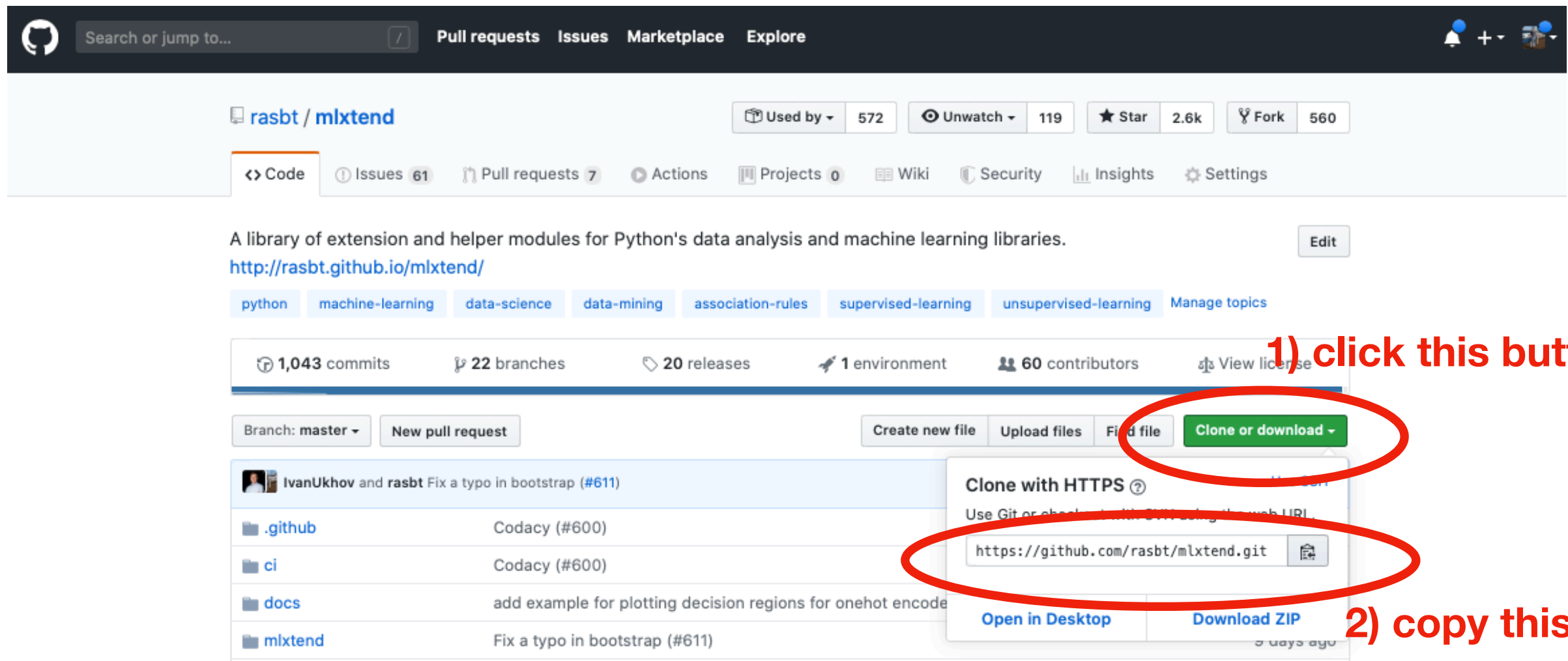
I would be very happy about any kind of contributions that help to improve and extend the functionality of mlxtend.

Quick Contributor Checklist

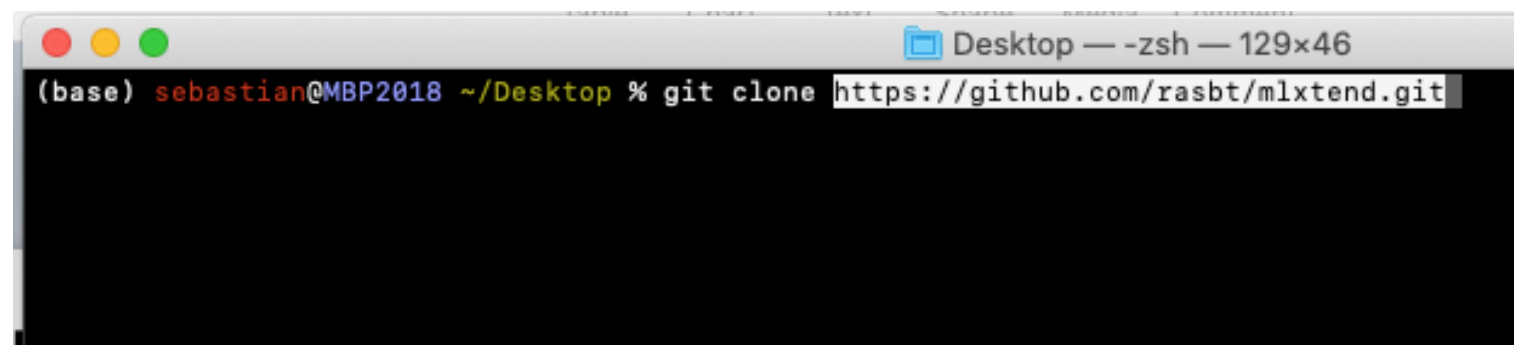
This is a quick checklist about the different steps of a typical contribution to mlxtend (and other open source projects). Consider copying this list to a local text file (or the issue

Basic Workflow for (most) Open Source Projects

Step 1: Clone the repository



The screenshot shows the GitHub repository page for 'rasbt/mlxtend'. The repository has 572 uses, 119 unwatchers, 2.6k stars, and 560 forks. The 'Code' tab is selected, showing the repository's description: 'A library of extension and helper modules for Python's data analysis and machine learning libraries.' The repository has 1,043 commits, 22 branches, 20 releases, 1 environment, and 60 contributors. The 'Clone or download' button is circled in red, with the annotation '1) click this button'. The 'Clone with HTTPS' dropdown menu is also circled in red, showing the URL 'https://github.com/rasbt/mlxtend.git', with the annotation '2) copy this link'.

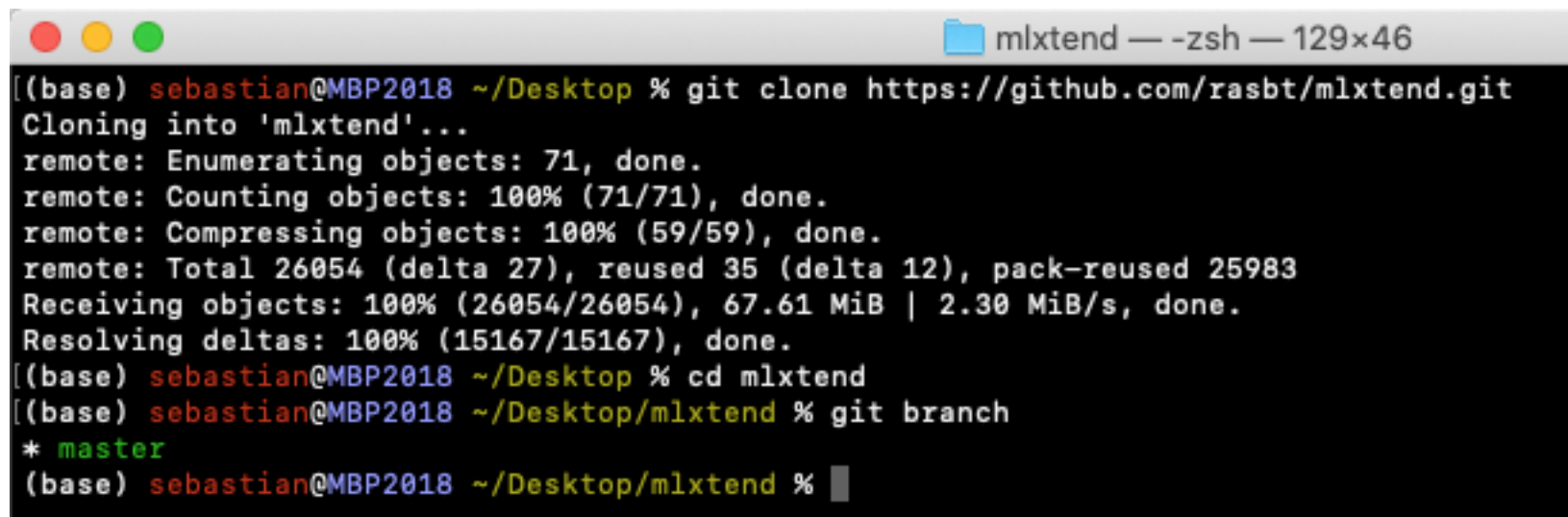


```
(base) sebastian@MBP2018 ~/Desktop % git clone https://github.com/rasbt/mlxtend.git
```


Basic Workflow for (most) Open Source Projects

Step 1: Clone the repository

4) Navigate to the cloned directory

A terminal window titled 'mlxtend — -zsh — 129x46' showing the process of cloning a repository. The user runs 'git clone https://github.com/rasbt/mlxtend.git' from the Desktop directory. The output shows the cloning progress, including object enumeration, counting, and compression. After cloning, the user runs 'cd mlxtend' to navigate into the directory and 'git branch' to check the current branch, which is 'master'.

```
[(base) sebastian@MBP2018 ~/Desktop % git clone https://github.com/rasbt/mlxtend.git
Cloning into 'mlxtend'...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (59/59), done.
remote: Total 26054 (delta 27), reused 35 (delta 12), pack-reused 25983
Receiving objects: 100% (26054/26054), 67.61 MiB | 2.30 MiB/s, done.
Resolving deltas: 100% (15167/15167), done.
[(base) sebastian@MBP2018 ~/Desktop % cd mlxtend
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % git branch
* master
(base) sebastian@MBP2018 ~/Desktop/mlxtend %
```

Step 2: Find an Issue to Work On

The screenshot shows the GitHub repository page for 'rasbt/mlxtend'. The 'Issues' tab is selected and highlighted with a red circle. A red arrow points to the 'Issues' tab with the text '1) click this button'. Below the repository header, there is a notification box about labeling issues for new contributors. The 'Filters' section shows 'is:issue is:open'. The 'Issues' list shows two issues, with the second issue, 'Rename 'per-class accuracy' to 'average per-class accuracy'', highlighted with a red circle. The issue is labeled 'easy' and 'stat479'.

1) click this button

Label issues and pull requests for new contributors
Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#)
[Go to Labels](#)

Filters Labels 17 Milestones 0 [New issue](#)

<input type="checkbox"/>	61 Open ✓ 205 Closed	Author	Labels	Projects	Milestones	Assignee	Sort
<input type="checkbox"/>	🔔 Standalone `average_per_class_accuracy` function easy stat479 #617 opened 2 hours ago by rasbt						
<input type="checkbox"/>	🔔 Rename 'per-class accuracy' to 'average per-class accuracy' easy stat479 #616 opened 2 hours ago by rasbt						

2) click on an issue that interests you

Step 2: Find an Issue to Work On

The screenshot shows a GitHub repository page for 'mlxtend' by 'rasbt'. The issue title is 'Rename 'per-class accuracy' to 'average per-class accuracy'' with ID #616. It has 119 watchers, 2.6k stars, and 560 forks. The issue is labeled 'easy' and 'stat479'. A comment by 'rasbt' is visible, and a new comment is being drafted.

Issue Title: Rename 'per-class accuracy' to 'average per-class accuracy' #616

Labels: easy, stat479

Comment by rasbt (2 hours ago):

In the `mlxtend.evaluate.scoring` function (http://rasbt.github.io/mlxtend/user_guide/evaluate/scoring/), the following renaming would make sense

- 'per-class accuracy' -> 'average per-class accuracy'
- 'per-class error' -> 'average per-class error'

to be more precise and avoid ambiguity (and potentially make room for related functions, related to #615).

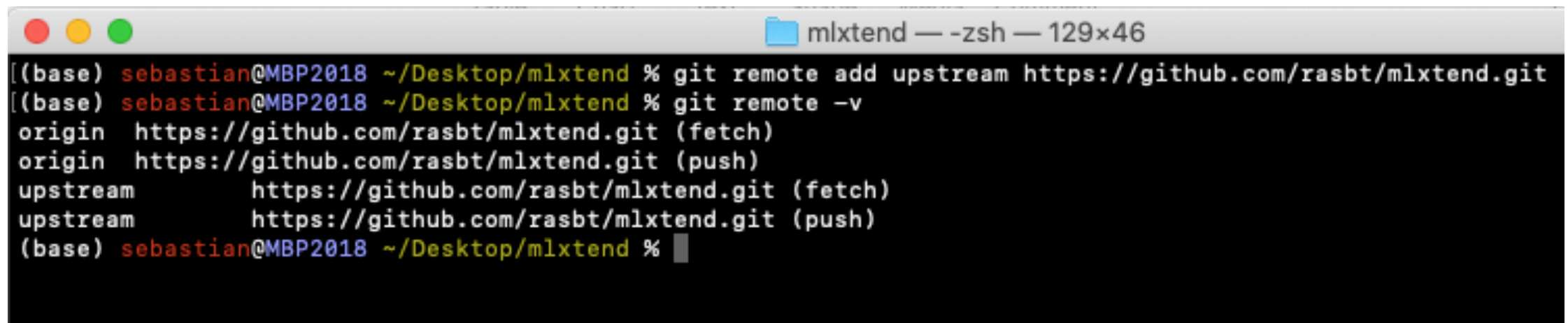
New Comment: I am working on this now ...

Buttons: Close and comment, Comment

3) add a comment to discuss the plan with the maintainer before working on it

Step 3: Keep Everything in Sync

- It's a good idea to periodically check that everything in your cloned directory is in sync with the original project on GitHub.
- For this, you have to setup an "upstream" link to the original repository (you only have to do it once after you cloned the repository!)

A terminal window titled 'mlxtend — -zsh — 129x46' showing a series of git commands and their output. The user is in a directory '~/Desktop/mlxtend'. The commands executed are: 'git remote add upstream https://github.com/rasbt/mlxtend.git' and 'git remote -v'. The output shows the 'origin' and 'upstream' remotes with their respective URLs and '(fetch)' and '(push)' capabilities.

```
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % git remote add upstream https://github.com/rasbt/mlxtend.git
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % git remote -v
origin  https://github.com/rasbt/mlxtend.git (fetch)
origin  https://github.com/rasbt/mlxtend.git (push)
upstream https://github.com/rasbt/mlxtend.git (fetch)
upstream https://github.com/rasbt/mlxtend.git (push)
(base) sebastian@MBP2018 ~/Desktop/mlxtend %
```

Step 3: Keep Everything in Sync

1) Before starting working on a new feature, fetch changes from the original repo

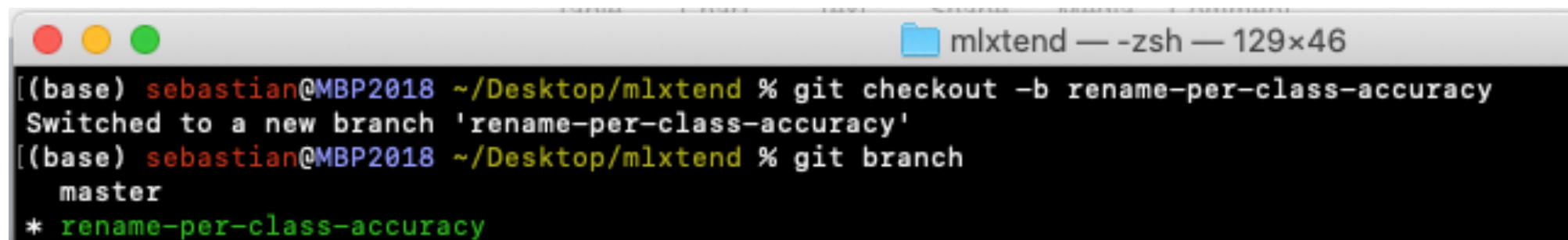
```
mlxtend — -zsh — 129x46
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git remote add upstream https://github.com/rasbt/mlxtend.git
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git remote -v
origin  https://github.com/rasbt/mlxtend.git (fetch)
origin  https://github.com/rasbt/mlxtend.git (push)
upstream https://github.com/rasbt/mlxtend.git (fetch)
upstream https://github.com/rasbt/mlxtend.git (push)
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git fetch upstream
```

2) Then merge the changes with the master branch

```
mlxtend — -zsh — 129x46
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git remote add upstream https://github.com/rasbt/mlxtend.git
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git remote -v
origin  https://github.com/rasbt/mlxtend.git (fetch)
origin  https://github.com/rasbt/mlxtend.git (push)
upstream https://github.com/rasbt/mlxtend.git (fetch)
upstream https://github.com/rasbt/mlxtend.git (push)
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git fetch upstream
From https://github.com/rasbt/mlxtend
 * [new branch]      adaline-docs      -> upstream/adaline-docs
 * [new branch]      categorical-decision -> upstream/categorical-decision
 * [new branch]      codacy            -> upstream/codacy
 * [new branch]      docfix            -> upstream/docfix
 * [new branch]      drop-proba        -> upstream/drop-proba
 * [new branch]      fix-feat-imp       -> upstream/fix-feat-imp
 * [new branch]      fixed-features     -> upstream/fixed-features
 * [new branch]      gh-pages          -> upstream/gh-pages
 * [new branch]      grid-support-feat-sele -> upstream/grid-support-feat-sele
 * [new branch]      heatmap           -> upstream/heatmap
 * [new branch]      ldaloadings        -> upstream/ldaloadings
 * [new branch]      master             -> upstream/master
 * [new branch]      mnist              -> upstream/mnist
 * [new branch]      pca-ddof           -> upstream/pca-ddof
 * [new branch]      pca-svd            -> upstream/pca-svd
 * [new branch]      py37               -> upstream/py37
 * [new branch]      readme-links       -> upstream/readme-links
 * [new branch]      sclf-warn          -> upstream/sclf-warn
 * [new branch]      setup              -> upstream/setup
 * [new branch]      stacking-dtype     -> upstream/stacking-dtype
 * [new branch]      v0.17.0            -> upstream/v0.17.0
 * [new branch]      verbose-apriori    -> upstream/verbose-apriori
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git merge upstream/master
Already up to date.
(base) sebastian@MBP2018 ~/Desktop/mlxtend %
```

Step 4: Make a New Feature Branch

- Now, create a new feature branch, which will be the branch where you will work on the new feature you are planning to implement or issue you are going to fix
- Then, execute "git branch" to ensure that you are on the correct branch now

A terminal window titled 'mlxtend — -zsh — 129x46' showing the execution of git commands. The user is in a directory '~/Desktop/mlxtend'. The first command is 'git checkout -b rename-per-class-accuracy', which results in the message 'Switched to a new branch 'rename-per-class-accuracy''. The second command is 'git branch', which lists the current branches: 'master' and '* rename-per-class-accuracy' (the active branch, indicated by an asterisk and green text).

```
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % git checkout -b rename-per-class-accuracy
Switched to a new branch 'rename-per-class-accuracy'
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % git branch
master
* rename-per-class-accuracy
```


Step 5: Start Coding

- You can start working on the feature/issue now

```
mlxtend — -zsh — 129x46
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % ls
LICENSE-BSD3.txt  MANIFEST.in      ci                files.txt         paper.bib         requirements.txt  setup.py
LICENSE-CC-BY.txt README.md         docs             mlxtend          paper.md          setup.cfg
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % code mlxtend/evaluate/scoring.py
```

```
scoring.py
Users > sebastian > Desktop > mlxtend > mlxtend > evaluate > scoring.py > scoring
22     scores = []
23     for l in unique_labels:
24         scores.append(func(np.where(true != l, 1, 0),
25                             np.where(pred != l, 1, 0)))
26     return float(sum(scores)) / len(scores)
27
28
29 def scoring(y_target, y_predicted, metric='error',
30             positive_label=1, unique_labels='auto'):
31     """Compute a scoring metric for supervised learning.
32
33     Parameters
34     -----
35     y_target : array-like, shape=[n_values]
36         True class labels or target values.
37     y_predicted : array-like, shape=[n_values]
38         Predicted class labels or target values.
39     metric : str (default: 'error')
40         Performance metric:
41         'accuracy': (TP + TN)/(FP + FN + TP + TN) = 1-ERR\n
42         'average per-class accuracy': Average per-class accuracy\n
43         'average per-class error': Average per-class error\n
```

Step 6: Code Testing

- After developing the new code, also add or modify the test functions

```
mlxtend — -zsh — 129x46
(base) sebastian@MBP2018 ~/Desktop/mlxtend % code mlxtend/evaluate/tests/test_scoring.py
```

```
test_scoring.py
scoring.py test_scoring.py
Users > sebastian > Desktop > mlxtend > mlxtend > evaluate > tests > test_scoring.py > test_avg_perclass_error

101
102 def test_f1():
103     y_targ = [1, 1, 1, 0, 0, 1, 0, 1]
104     y_pred = [1, 0, 1, 0, 0, 0, 1, 1]
105     res = scoring(y_target=y_targ, y_predicted=y_pred, metric='f1')
106     assert round(res, 3) == 0.667, res
107
108
109 def test_matthews_corr_coef():
110     y_targ = [1, 1, 1, 0, 0, 1, 0, 1]
111     y_pred = [1, 0, 1, 0, 0, 0, 1, 1]
112     res = scoring(y_target=y_targ,
113                  y_predicted=y_pred,
114                  metric='matthews_corr_coef')
115     assert round(res, 3) == 0.258, res
116
117
118 def test_avg_perclass_accuracy():
119     y_targ = np.array([0, 0, 0, 1, 1, 1, 1, 1, 2, 2])
120     y_pred = np.array([0, 1, 1, 0, 1, 1, 2, 2, 2, 2])
121     res = scoring(y_target=y_targ,
122                  y_predicted=y_pred,
123                  metric='average per-class accuracy')
124     assert round(res, 3) == 0.667, res
125
126
127 def test_avg_perclass_error():
128     y_targ = np.array([0, 0, 0, 1, 1, 1, 1, 1, 2, 2])
129     y_pred = np.array([0, 1, 1, 0, 1, 1, 2, 2, 2, 2])
130     res = scoring(y_target=y_targ,
131                  y_predicted=y_pred,
132                  metric='average per-class error')
133     assert round(res, 3) == 0.333, res
134
```


Step 6: Code Testing

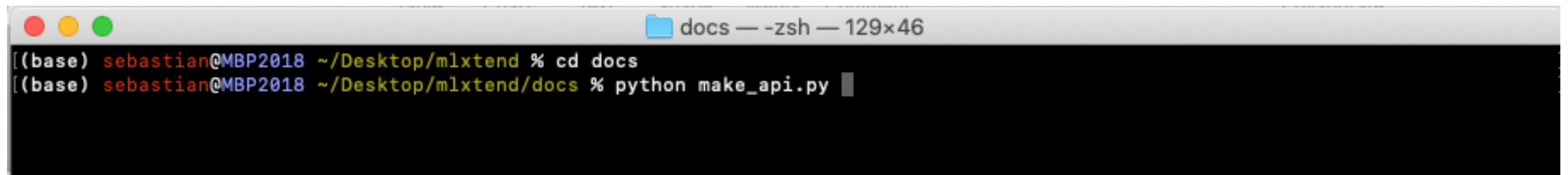
- Then, test your code locally to make sure that your changes didn't break the existing code base

```
mlxtend — -zsh — 129x46
(base) sebastian@MBP2018 ~/Desktop/mlxtend % PYTHONPATH='.' pytest ./mlxtend -sv
```

```
mlxtend — -zsh — 129x46
mlxtend/regressor/tests/test_stacking_cv_regression.py::test_weight_unsupported_with_no_weight PASSED
mlxtend/regressor/tests/test_stacking_cv_regression.py::test_gridsearch_replace_mix PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_different_models PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_multivariate PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_multivariate_class PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_sample_weight PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_weight_ones PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_weight_unsupported_regressor PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_weight_unsupported_meta PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_weight_unsupported_with_no_weight PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_gridsearch PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_gridsearch_numerate_regr PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_get_coeff PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_get_intercept PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_get_coeff_fail PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_get_params PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_regressor_gridsearch PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_predict_meta_features PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_train_meta_features PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_not_fitted_predict PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_clone PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_features_in_secondary 0.13732094333079276
0.12403094909404185
PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_predictions_from_sparse_matrix 0.6082588436138667
0.6082588436138667
PASSED
mlxtend/regressor/tests/test_stacking_regression.py::test_sparse_matrix_inputs_and_features_in_secondary PASSED
mlxtend/text/tests/test_generalize_names.py::test_generalize_names PASSED
mlxtend/text/tests/test_generalize_names_duplcheck.py::test_generalize_names_duplcheck PASSED
mlxtend/text/tests/test_tokenizer.py::test_tokenizer_words_and_emoticons_1 PASSED
mlxtend/text/tests/test_tokenizer.py::test_tokenizer_words_and_emoticons_2 PASSED
mlxtend/utils/tests/test_checking_inputs.py::test_check_Xy_ok PASSED
mlxtend/utils/tests/test_checking_inputs.py::test_check_Xy_invalid_type_X PASSED
mlxtend/utils/tests/test_checking_inputs.py::test_check_Xy_float16_X PASSED
mlxtend/utils/tests/test_checking_inputs.py::test_check_Xy_float16_y PASSED
mlxtend/utils/tests/test_checking_inputs.py::test_check_Xy_invalid_type_y PASSED
mlxtend/utils/tests/test_checking_inputs.py::test_check_Xy_invalid_dtype_X PASSED
mlxtend/utils/tests/test_checking_inputs.py::test_check_Xy_invalid_dtype_y PASSED
```

Step 7: Update the Documentation

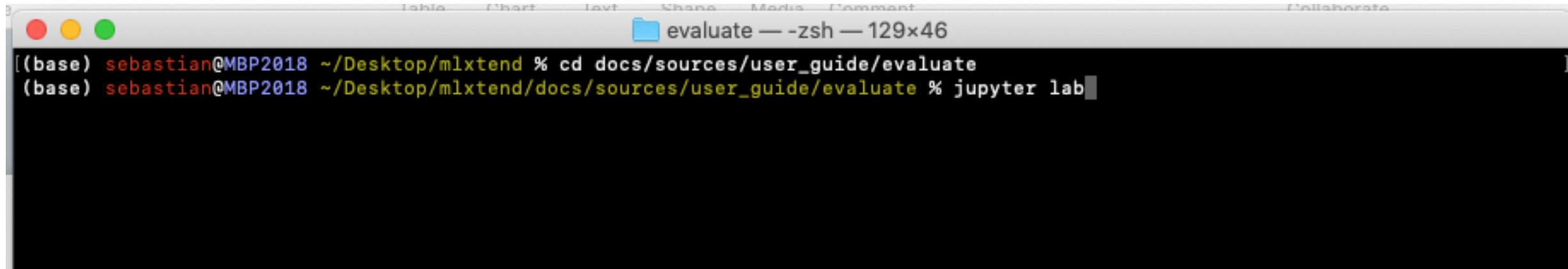
- Run the python make_api.py file to update the general code documentation (this is mlxtend-specific)



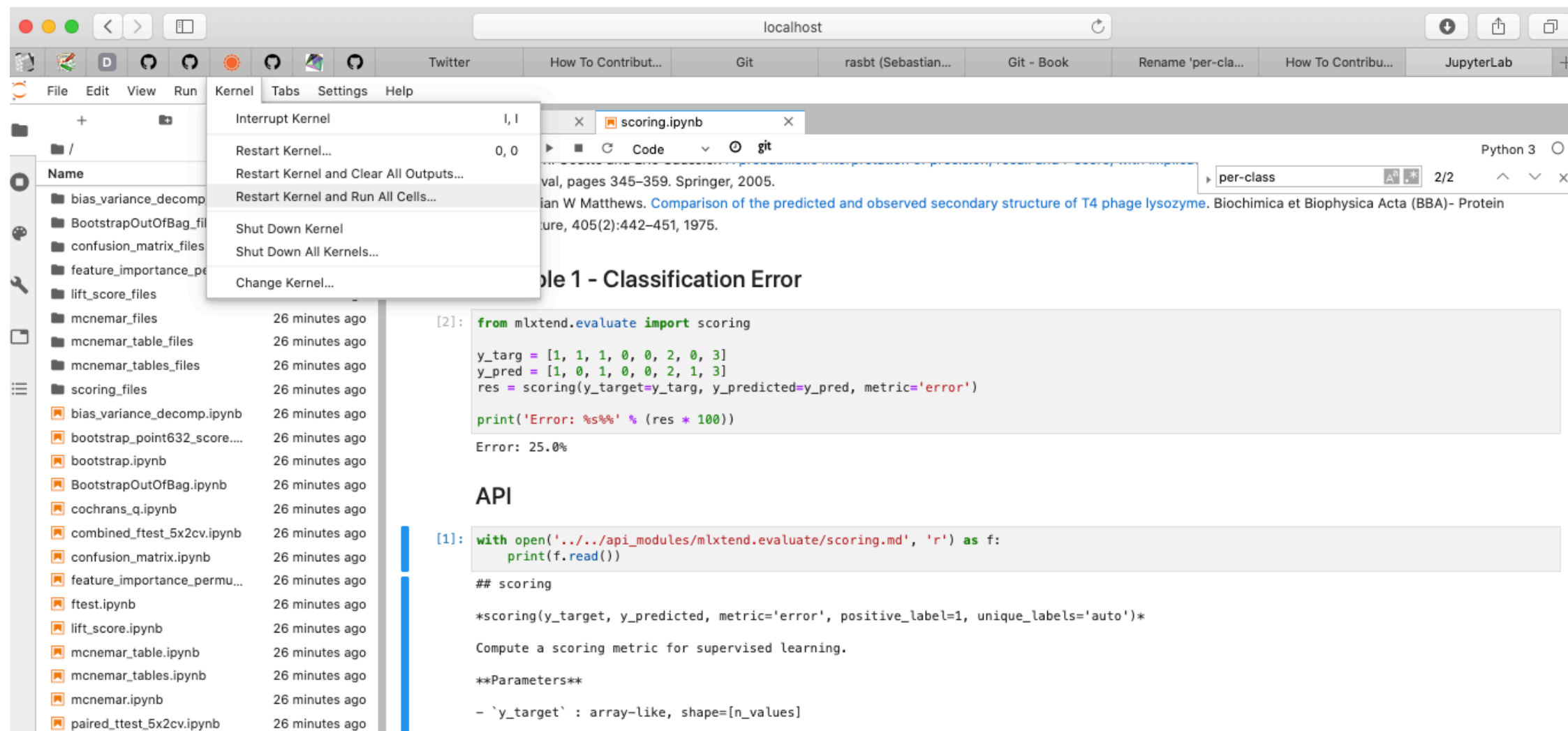
```
docs — -zsh — 129x46
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % cd docs
[(base) sebastian@MBP2018 ~/Desktop/mlxtend/docs % python make_api.py
```

Step 7: Update the Documentation

- Next, update the documentation on the Jupyter notebook (this is mlxtend-specific)



```
evaluate — zsh — 129x46
(base) sebastian@MBP2018 ~/Desktop/mlxtend % cd docs/sources/user_guide/evaluate
(base) sebastian@MBP2018 ~/Desktop/mlxtend/docs/sources/user_guide/evaluate % jupyter lab
```



Step 8: Make a Changelog Entry

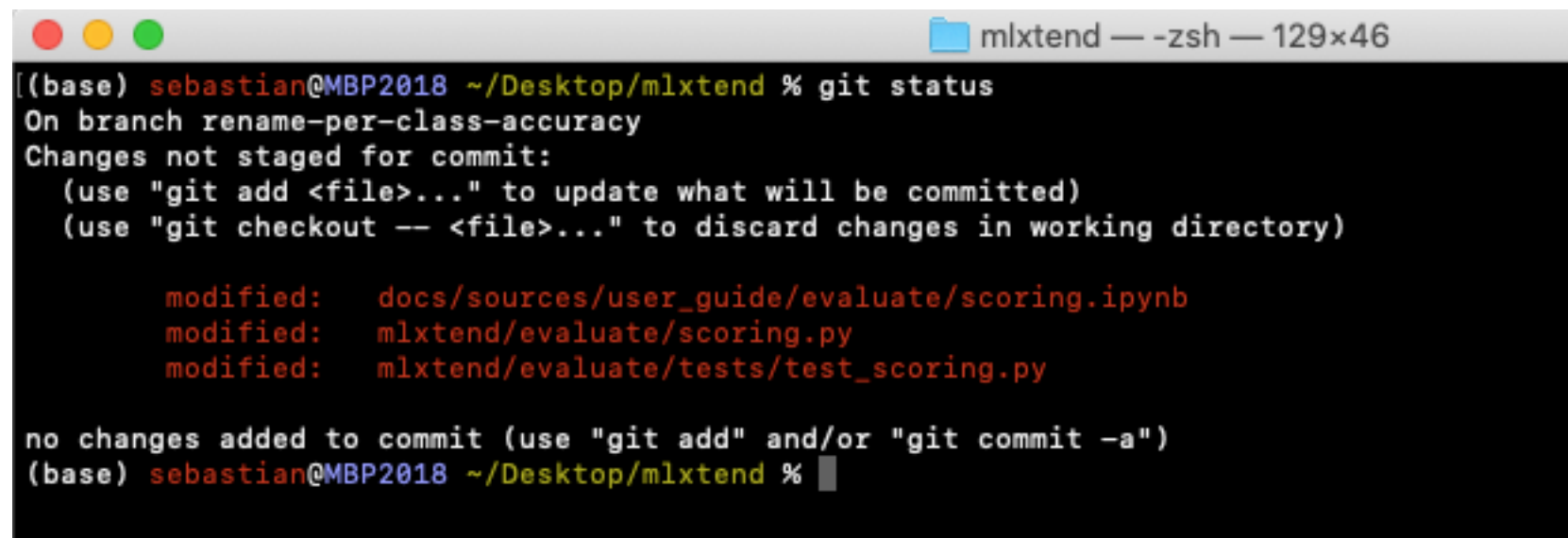
```
mlxtend — -zsh — 129x46
((base) sebastian@MBP2018 ~/Desktop/mlxtend % code docs/sources/CHANGELOG.md
```

```
CHANGELOG.md
scoring.py test_scoring.py CHANGELOG.md
Users > sebastian > Desktop > mlxtend > docs > sources > CHANGELOG.md > abc # Release Notes > abc ### Version 0.18.0 (TBD) > abc ##### Changes
> per-class Aa AbI .* 1 of 4 ↑ ↓ ≡ ×

1 # Release Notes
2
3 ---
4
5 The CHANGELOG for the current development version is available at
6 [https://github.com/rasbt/mlxtend/blob/master/docs/sources/CHANGELOG.md] (https://github.com/rasbt/mlxtend/blob/master/docs/sources/CHANGELOG.md).
7
8 ---
9
10 ### Version 0.18.0 (TBD)
11
12 ##### Downloads
13
14 - [Source code (zip)](https://github.com/rasbt/mlxtend/archive/v0.18.0.zip)
15
16 - [Source code (tar.gz)](https://github.com/rasbt/mlxtend/archive/v0.18.0.tar.gz)
17
18 ##### New Features
19
20 - The SequentialFeatureSelector now supports using pre-specified feature sets via the fixed_features parameter. ([#578])(https://github.com/rasbt/mlxtend/pull/578)
21
22 ##### Changes
23
24 - Changed 'per-class accuracy' and 'per-class error' to 'average 'per-class accuracy' and 'average 'per-class error', respectively, to improve clarity.
25
```

Step 9: Upload your Changes

- Use "git status" to see what files you have changes

A terminal window titled 'mlxtend — -zsh — 129x46' showing the output of the 'git status' command. The output indicates that the user is on the 'rename-per-class-accuracy' branch and lists three modified files: 'docs/sources/user_guide/evaluate/scoring.ipynb', 'mlxtend/evaluate/scoring.py', and 'mlxtend/evaluate/tests/test_scoring.py'. It also provides instructions on how to stage changes with 'git add' and how to discard changes with 'git checkout --'.

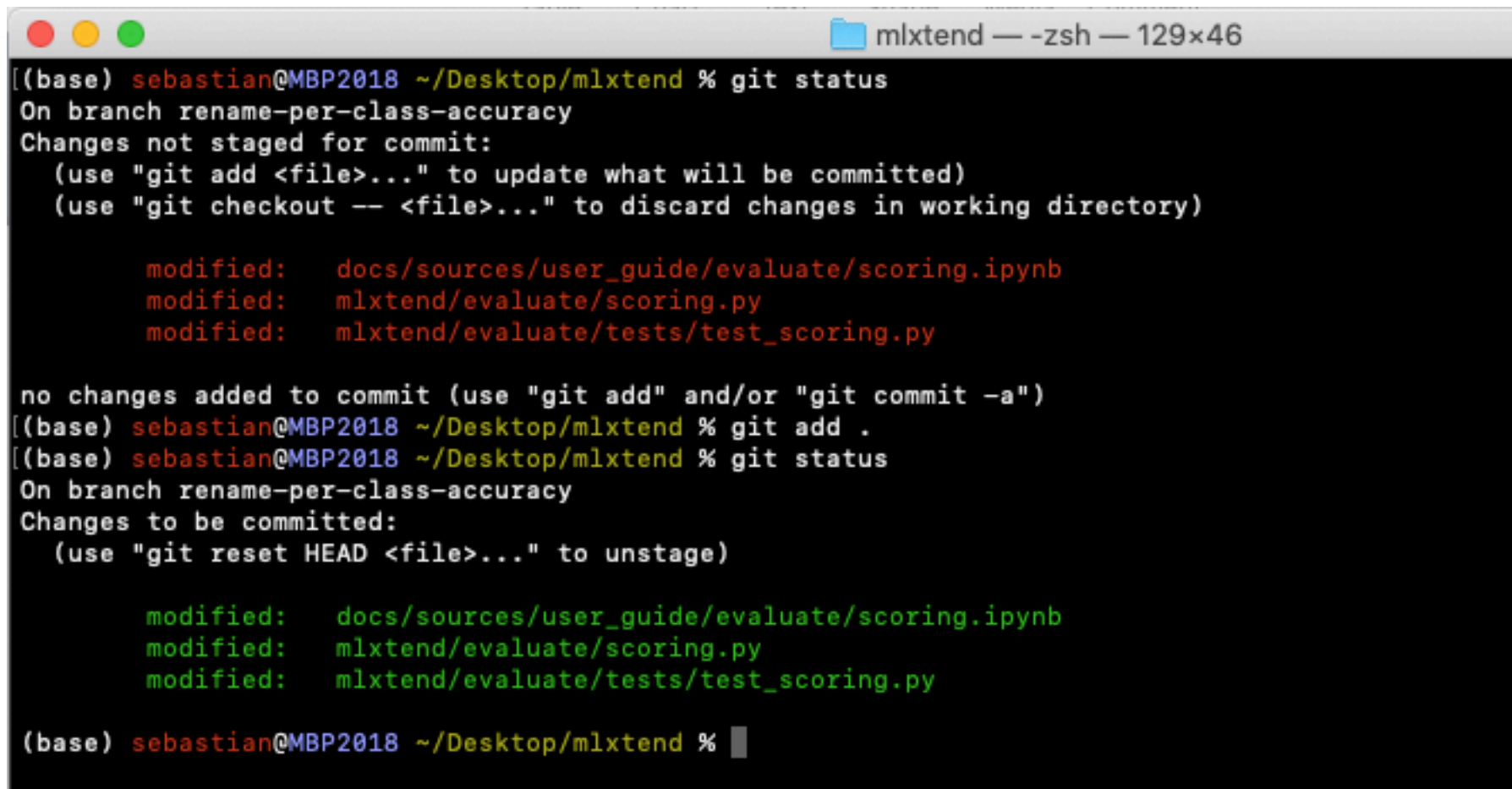
```
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % git status
On branch rename-per-class-accuracy
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   docs/sources/user_guide/evaluate/scoring.ipynb
        modified:   mlxtend/evaluate/scoring.py
        modified:   mlxtend/evaluate/tests/test_scoring.py

no changes added to commit (use "git add" and/or "git commit -a")
(base) sebastian@MBP2018 ~/Desktop/mlxtend %
```

Step 9: Upload your Changes

- Then add the files you want to upload via "git add filename"
- If the "git status" list does not include any files that you don't want to upload, you can also execute "git add ." to stage all files for uploading all files

A terminal window titled 'mlxtend — -zsh — 129x46' showing the execution of git commands. The user is in a directory named 'mlxtend' on a branch named 'rename-per-class-accuracy'. The first 'git status' command shows three modified files. The user then runs 'git add .' to stage all changes. A second 'git status' command shows the same three files are now staged for commit.

```
((base) sebastian@MBP2018 ~/Desktop/mlxtend % git status
On branch rename-per-class-accuracy
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   docs/sources/user_guide/evaluate/scoring.ipynb
        modified:   mlxtend/evaluate/scoring.py
        modified:   mlxtend/evaluate/tests/test_scoring.py

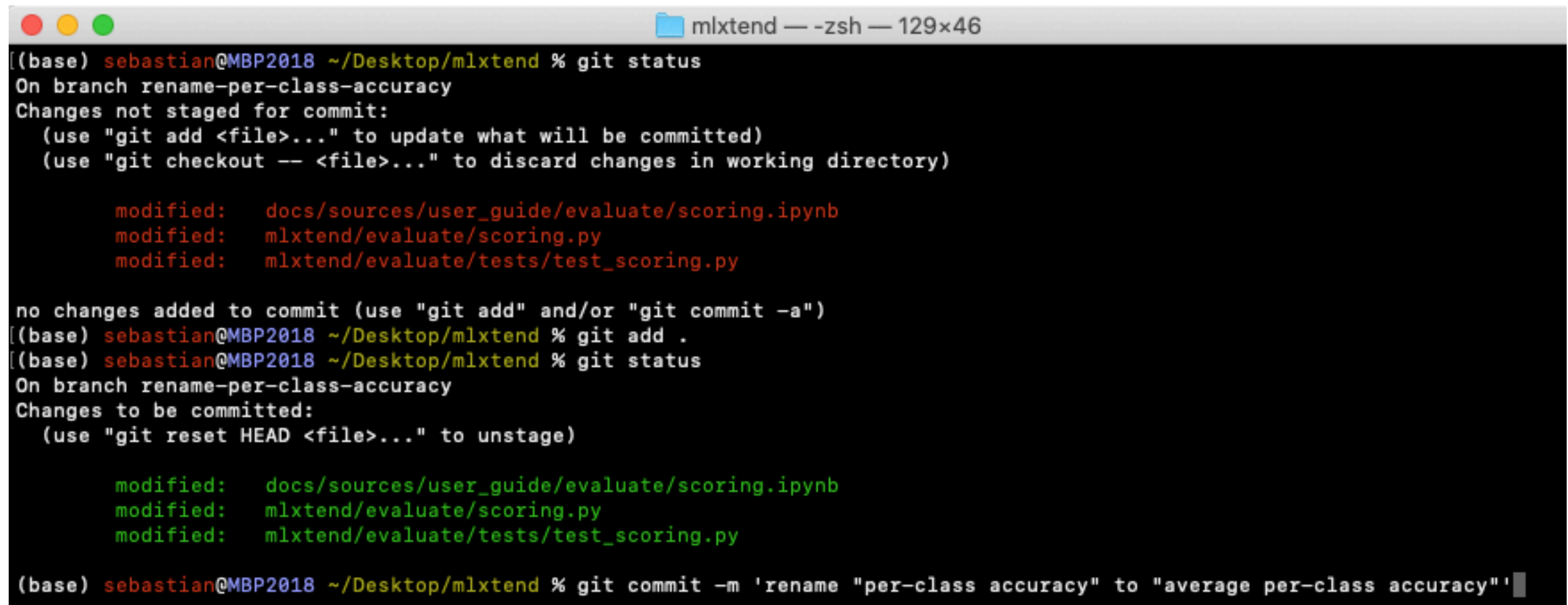
no changes added to commit (use "git add" and/or "git commit -a")
((base) sebastian@MBP2018 ~/Desktop/mlxtend % git add .
((base) sebastian@MBP2018 ~/Desktop/mlxtend % git status
On branch rename-per-class-accuracy
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   docs/sources/user_guide/evaluate/scoring.ipynb
        modified:   mlxtend/evaluate/scoring.py
        modified:   mlxtend/evaluate/tests/test_scoring.py

(base) sebastian@MBP2018 ~/Desktop/mlxtend %
```


Step 9: Upload your Changes

- After marking the files you want to upload via "git add", you can now use "git commit -m 'your message'" to make a entry in the GitHub history

A terminal window titled 'mlxtend — -zsh — 129x46' showing the execution of git commands. The user is in a directory named 'mlxtend' on a branch named 'rename-per-class-accuracy'. The first 'git status' command shows three modified files. The user then runs 'git add .' to stage these changes. A second 'git status' command shows the same three files as 'Changes to be committed'. Finally, the user runs 'git commit -m 'rename "per-class accuracy" to "average per-class accuracy"' to create a new commit.

```
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % git status
On branch rename-per-class-accuracy
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   docs/sources/user_guide/evaluate/scoring.ipynb
        modified:   mlxtend/evaluate/scoring.py
        modified:   mlxtend/evaluate/tests/test_scoring.py

no changes added to commit (use "git add" and/or "git commit -a")
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % git add .
[(base) sebastian@MBP2018 ~/Desktop/mlxtend % git status
On branch rename-per-class-accuracy
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   docs/sources/user_guide/evaluate/scoring.ipynb
        modified:   mlxtend/evaluate/scoring.py
        modified:   mlxtend/evaluate/tests/test_scoring.py

(base) sebastian@MBP2018 ~/Desktop/mlxtend % git commit -m 'rename "per-class accuracy" to "average per-class accuracy"'
```

Step 9: Upload your Changes

- Push your changes to the server

```
mlxtend — -zsh — 129x46
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git status
On branch rename-per-class-accuracy
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   docs/sources/user_guide/evaluate/scoring.ipynb
        modified:   mlxtend/evaluate/scoring.py
        modified:   mlxtend/evaluate/tests/test_scoring.py

no changes added to commit (use "git add" and/or "git commit -a")
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git add .
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git status
On branch rename-per-class-accuracy
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   docs/sources/user_guide/evaluate/scoring.ipynb
        modified:   mlxtend/evaluate/scoring.py
        modified:   mlxtend/evaluate/tests/test_scoring.py

(base) sebastian@MBP2018 ~/Desktop/mlxtend % git commit -m 'rename "per-class accuracy" to "average per-class accuracy"'
[rename-per-class-accuracy 182d81f1] rename "per-class accuracy" to "average per-class accuracy"
3 files changed, 14 insertions(+), 16 deletions(-)
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git status
On branch rename-per-class-accuracy
nothing to commit, working tree clean
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git branch
* master
* rename-per-class-accuracy
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git push origin rename-per-class-accuracy

master
* rename-per-class-accuracy
(base) sebastian@MBP2018 ~/Desktop/mlxtend % git push origin rename-per-class-accuracy
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 12 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 986 bytes | 986.00 KiB/s, done.
Total 12 (delta 11), reused 0 (delta 0)
remote: Resolving deltas: 100% (11/11), completed with 11 local objects.
remote:
remote: Create a pull request for 'rename-per-class-accuracy' on GitHub by visiting:
remote:   https://github.com/rasbt/mlxtend/pull/new/rename-per-class-accuracy
remote:
To https://github.com/rasbt/mlxtend.git
* [new branch]      rename-per-class-accuracy -> rename-per-class-accuracy
(base) sebastian@MBP2018 ~/Desktop/mlxtend %
```


Step 10: Open a Pull Request

The screenshot shows the GitHub repository page for 'rasbt/mlxtend'. The repository description is 'A library of extension and helper modules for Python's data analysis and machine learning libraries.' with a link to 'http://rasbt.github.io/mlxtend/'. The repository has 1,043 commits, 23 branches, 20 releases, 1 environment, 60 contributors, and 2.6k stars. The 'Compare & pull request' button is circled in red, with a red arrow pointing to it from the text '1) click this button'.

Search or jump to... / Pull requests Issues Marketplace Explore

rasbt / mlxtend

Unwatch 119 Star 2.6k Fork 560

Code Issues 61 Pull requests 7 Actions Projects 0 Wiki Security Insights Settings

A library of extension and helper modules for Python's data analysis and machine learning libraries. <http://rasbt.github.io/mlxtend/> Edit

python machine-learning data-science data-mining association-rules supervised-learning unsupervised-learning Manage topics

1,043 commits 23 branches 20 releases 1 environment 60 contributors View license

Your recently pushed branches:

rename-per-class-accuracy (less than a minute ago) **Compare & pull request**

Branch: master New pull request Create new file Upload files Find file Clone or download

IvanUkhov and rasbt Fix a typo in bootstrap (#611) Latest commit c53586a 9 days ago

.github	Codacy (#600)	last month
ci	Codacy (#600)	last month
docs	add example for plotting decision regions for onehot encoded labels (#...	15 days ago
mlxtend	Fix a typo in bootstrap (#611)	9 days ago
.appveyor.yml	Use native joblib instead of sklearn.externals.joblib (#548)	5 months ago
.coveragerc	updating six and adding externals	4 years ago
.gitignore	Adding PCA correlation circle graph (#544)	5 months ago
travis.yml	und travis from python 3.6 to python 3.8 (#600)	13 days ago

Step 10: Open a Pull Request

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master ← compare: rename-per-class-accuracy ✓ **Able to merge.** These branches can be automatically merged.

rename "per-class accuracy" to "average per-class accuracy"

Write Preview

Description

- Changed `'per-class accuracy'` and `'per-class error'` to `'average per-class accuracy'` and `'average per-class error'`, respectively, to improve clarity.

Related issues or pull requests

<!--
If applicable, please link related issues/pull request here. E.g.,
Fixes #366
-->

Fixes #616

Pull Request Checklist

- [] Added a note about the modification or contribution to the `./docs/sources/CHANGELOG.md` file (if applicable)
- [] Added appropriate unit test functions in the `./mlxtend/*/tests` directories (if applicable)
- [] Modify documentation in the corresponding Jupyter Notebook under `mlxtend/docs/sources/` (if applicable)
- [] Ran `PYTHONPATH='.' pytest ./mlxtend -sv` and make sure that all unit tests pass (for small modifications, it might be sufficient to only run the specific test file, e.g., `PYTHONPATH='.' pytest ./mlxtend/classifier/tests/test_stacking_cv_classifier.py -sv`)
- [] Checked for style issues by running `flake8 ./mlxtend`

Attach files by dragging & dropping, selecting or pasting them.

2) fill out the template that will be provided

3) click this button

Create pull request

Step 10: Open a Pull Request

rename "per-class accuracy" to "average per-class accuracy" #618

 Open rasbt wants to merge 1 commit into `master` from `rename-per-class-accuracy`

 Conversation 0  Commits 1  Checks 0  Files changed 3

rasbt commented 18 seconds ago • edited • Owner + 👤 ...

Description


- Changed 'per-class accuracy' and 'per-class error' to average 'per-class accuracy' and average 'per-class error', respectively, to improve clarity.

Related issues or pull requests

Fixes #616

Pull Request Checklist


- ☒ Added a note about the modification or contribution to the `./docs/sources/CHANGELOG.md` file (if applicable)
- ☒ Added appropriate unit test functions in the `./mlxtend/*/tests` directories (if applicable)
- ☒ Modify documentation in the corresponding Jupyter Notebook under `mlxtend/docs/sources/` (if applicable)
- ☒ Ran `PYTHONPATH='.' pytest ./mlxtend -sv` and make sure that all unit tests pass (for small modifications, it might be sufficient to only run the specific test file, e.g., `PYTHONPATH='.' pytest ./mlxtend/classifier/tests/test_stacking_cv_classifier.py -sv`)
- ☒ Checked for style issues by running `flake8 ./mlxtend`




 rename "per-class accuracy" to "average per-class accuracy" 182d81f

4) check off these items

5) check the automated unit test results later

The branch has not been deployed
No deployments

 **Some checks haven't completed yet** [Hide all checks](#)
3 pending checks

	Codacy/PR Quality Review Pending — Hang in there, Codacy is reviewin...	Details
	continuous-integration/appveyor/branch Pending — Waiting for AppVe...	Details
	continuous-integration/travis-ci/push Pending — The Travis CI build is ...	Details