W4995 Applied Machine Learning

Tools and infrastructure

01/23/17

Andreas Müller

So you think you know git?


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.


Discard usability
Obtain arduousness

git Basics

- Repository

  $ git init

  $ rm .git

- Commit

- Remote

git Basics

- Repository

  $ git init

  $ rm .git

- Commit

- Remote

Typical Workflow

- Clone
- Branch
- Add, Commit, add add, commit, add commit
- Merge
- push

Some tips

- Git status

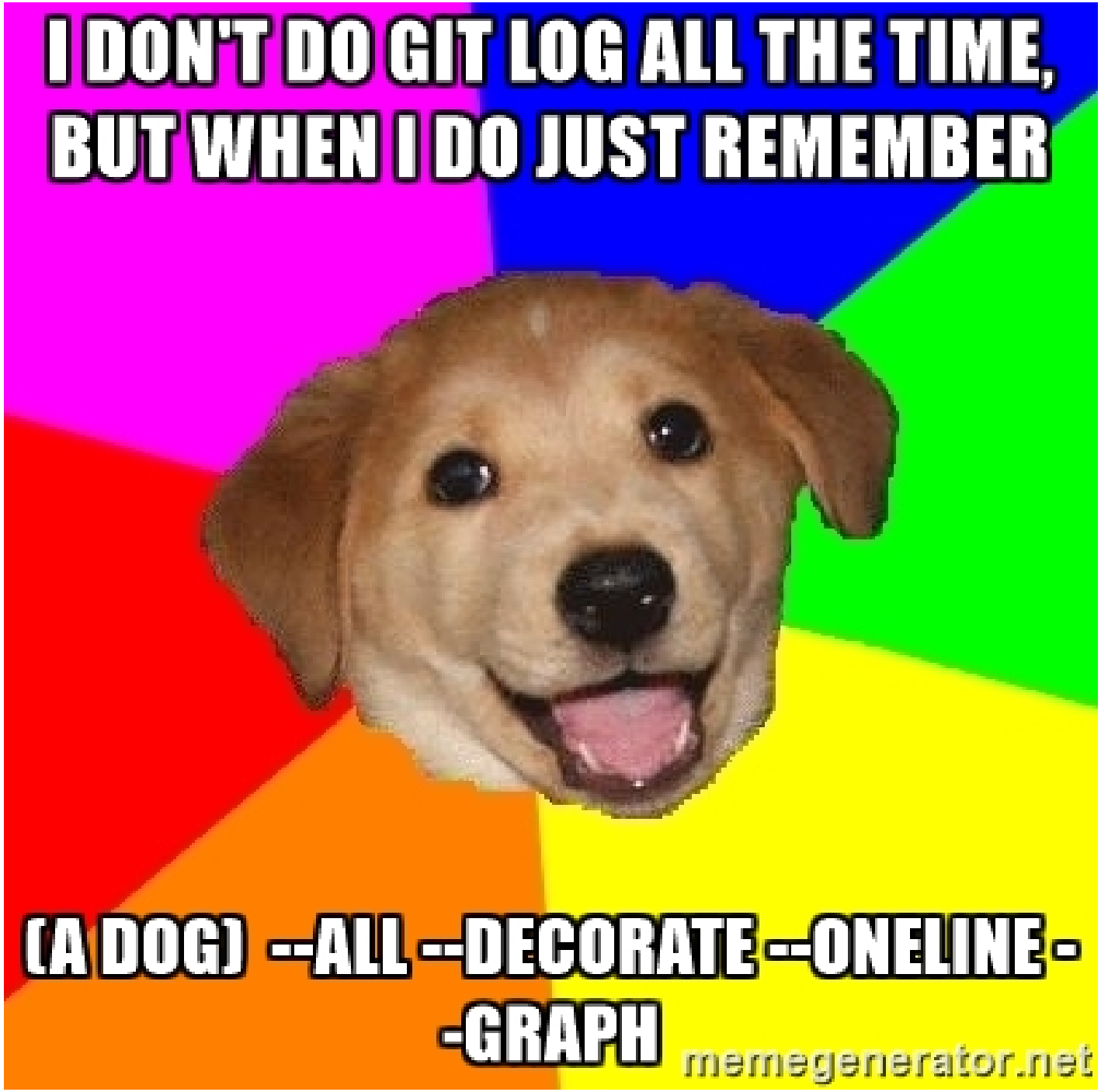
- Install shell plugins to show branch (oh-my-zsh)

```
/home/andy/checkout/scikit-learn [git::master *] [andy@dsi-amueller] [16:35]
>
```

- Set your editor, pager and diff-tool

Git log

```
/home/andy/checkout/scikit-learn [git::master *] [andy@dsi-amueller] [16:43]
> git log --oneline --decorate --all --graph -n 50
* 9616acf (HEAD -> master, upstream/master) CI remove obsolete comment
* 7978119 [MRG] #8218: in FAQ, link deep learning question to GPU question (#8220)
* be305ce TST/FIX Add check for estimator: parameters not modified by `fit` (#7846)
* aaebee1 FIX Issue #8173 - pass n_neighbors in MI computation (#8181)
* 4826883 Call sorted on lfw folder path contents (#7648)
* 4907029 [MRG+3] FIX Memory leak in MAE; Use safe_realloc; Acquire GIL only when raising; Propagate all errors to python interpreter level (#7811) (#80
* 08772c4 FIX Ensure coef_ is an ndarray when fitting LassoLars (#8160)
* 5d6460d MNT/BLD Use GitHub's merge refs to test PRs on CircleCI (#8211)
* 66443aa [MRG+1] Add prominent mention of Laplacian Eigenmaps (#8155)
* bddda7b [MRG + 2] [MAINT] Update to Sphinx-Gallery 0.1.7 (#7986)
* aea6462 [MRG+1] Fixes #8198 - error in datasets.make_moons (#8199)
* 0eb33ad TRAVIS fix flake8_diff.sh check_files (#8208)
* eabaef3 DOC add missing parentheses in TfidfTrasnformer docstring
* 3dc8d2f DOC additional fixes to 20 newsgroups to prevent TypeError (#8204)
* cbddb92 removed stray space in '__main__ ' (#8203)
* ca687ba Upgrade html documentation to jQuery v3.1.1 (#8145)
* 84cc67b Clarify error message for min_samples_split. (#8167)
* 2a1408a fixing typo in cs_mse_path_ deprecation (#8176)
* fd84a56 [MRG + 1] Fix the cross_val_predict function for method='predict_proba' (#7889)
* 4910e11 DOC Fix link (#8171)
* 8998856 Fix Ridge floating point instability (#8154)
* 2f7f5a1 [MRG + 1] Add fowlkess-mallows and other supervised cluster metrics to SCORERS dict so it can be used in hyper-param search (#8117)
* 8695ff5 [MRG + 1] add partial_fit to multioutput module (#8054)
* 0b02125 [MRG] FIX Avoid default mutable argument in constructor of AgglomerativeClustering (#8153)
* d0ce0d9 [MRG+2] Avoid failure in first iteration of RANSAC regression (#7914)
* e874398 [MRG+1] DOC: complete list of online learners (#8152)
* e0c60fe FIX sphinx gallery rendering of plot_digits_pipe example
* 84349a7 [MRG+1] Deprecate ridge_alpha param on SparsePCA.transform() (#8137)
* c49ced9 DOC: updating GridSearchCV's n_jobs parameter (#8106)
* 2cb7e47 [MRG+1] fowlkes_mallows_score: more unit tests (Fixes #8101) (#8140)
* 543b056 [MRG+1] Add DBSCAN support for additional metric params (#8139)
* d7e77ce [MRG+1] Fix "cite us" link in sidebar (#8142)
* 288827b [MRG] update copyright years for 2017 (#8138)
* e2adbb7 DOC Fix typo in FAQ (#8132)
* 986a49b FIX Split data using _safe_split in _permutaion_test_score (#5697)
* ab1c4d4 [MRG+1] Catch cases for different class size in MLPClassifier with warm start (#7976)  (#8035)
| * dcf24b9 (origin/repr_give_up, repr_give_up) i have no idea what I'm doing
| * b25fa4b pep8
| * 8b30325 playing around, then giving up
| | * 0d1398a (upstream/ignore_lambda_to_diff_errors) MNT Ignore E731: Use a def instead of lambda
| |/
|/|
* | d97d13e DOC add sklearn-crfsuite to related projects (#7878)
* | fcb706a [MRG+3] Fused types for MultiTaskElasticNet (#8061)
* | 096a9cb [MRG] MAINT Python 3.6 fixes (#8123)
* | e088685 DOC Fix indentation errors and username links (#8121)
* | 4f3c60c [MRG+2] FIX IsolationForest(max_features=0.8).predict(X) fails input validation (#5757)
```


I DON'T DO GIT LOG ALL THE TIME,
BUT WHEN I DO JUST REMEMBER
(A DOG) --ALL --DECORATE --ONELINE --GRAPH
memegenerator.net

Understanding Git!

- Working directory
- Index
- Staging area
- Branches
- Head

Commands – And what they do

- Git add
  puts files from working director into staging area. If not in index, adds them to tracked files.
- Git commit
  commits files from staging area to index
- Git checkout [<commit>] [<file>]
  Set <file> in working directory to state at <commit> *and stages it*.
- Git checkout [-b] <branch>
  moves HEAD to <branch> (-b creates it)
- Git reset --soft <commit>
  moves HEAD to <commit> (takes the current branch with it)
- Git reset --mixed <commit>
  moves HEAD to <commit>, changes index to be at <commit> (but not working directory)
- Git reset --hard <commit>
  moved HEAD to <commit>, changes index and working tree to <commit>

Merge

- Fast-forward merge:

```
* 6cec4ed (HEAD -> another_one) E
* 1e96a3b D
* 513ced1 (master) C
* b5ba00a B
* db928a0 A
```

```
/tmp/git_graphs [git::master] [andy@dsi-amueller] [15:38]
> git merge another_one
Updating 513ced1..6cec4ed
Fast-forward
 D | 0
 E | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 D
 create mode 100644 E
```

```
* 6cec4ed (HEAD -> master, another_one) E
* 1e96a3b D
* 513ced1 C
* b5ba00a B
* db928a0 A
```

- Merge-commits:

```
* 43563a5 (HEAD -> master) G
* c3ea8c8 F
| * 6cec4ed (another_one) E
| * 1e96a3b D
|/
* 513ced1 C
* b5ba00a B
* db928a0 A
```

```
/tmp/git_graphs [git::master] [andy@dsi-amueller] [15:43]
> git merge another_one
Merge made by the 'recursive' strategy.
 D | 0
 E | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 D
 create mode 100644 E
```

```
*   d6fedb0 (HEAD -> master) Merge branch 'another_one'
|\
| * 6cec4ed (another_one) E
| * 1e96a3b D
* | 43563a5 G
* | c3ea8c8 F
|/
* 513ced1 C
* b5ba00a B
* db928a0 A
```

Rebase

- Rebase

```
                A---B---C topic
               /
        D---E---F---G master

From this point, the result of either of the following commands:

    git rebase master
    git rebase master topic

would be:

                        A'--B'--C' topic
                       /
        D---E---F---G master
```

- Rebase onto:

```
o---o---o---o---o  master
     \
      o---o---o---o---o  next
                       \
                        o---o---o  topic
```

```
o---o---o---o---o  master
    |            \
    |             o'--o'--o'  topic
     \
      o---o---o---o---o  next
```

```
git rebase --onto master next
```

Squash before rebase!

Try to “squash” commits before doing a rebase – it might save you lots of conflict resolution!

Interactive rebase

```
git rebase -i <commit>
```

```
pick bddda7b [MRG + 2] [MAINT] Update to Sphinx-Gallery 0.1.7 (#7986)
pick 66443aa [MRG+1] Add prominent mention of Laplacian Eigenmaps (#8155)
pick 5d6460d MNT/BLD Use GitHub's merge refs to test PRs on CircleCI (#8211)
pick 08772c4 FIX Ensure coef_ is an ndarray when fitting LassoLars (#8160)
pick 4907029 [MRG+3] FIX Memory leak in MAE; Use safe_realloc; Acquire GIL only when rai
pick 4826883 Call sorted on lfw folder path contents (#7648)
pick aaebee1 FIX Issue #8173 - pass n_neighbors in MI computation (#8181)
pick be305ce TST/FIX Add check for estimator: parameters not modified by `fit` (#7846)
pick 7978119 [MRG] #8218: in FAQ, link deep learning question to GPU question (#8220)
pick 9616acf CI remove obsolete comment

# Rebase aea6462..9616acf onto aea6462 (10 command(s))
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
# d, drop = remove commit
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
```

Interactive adding

```
> git add -i
           staged     unstaged path
  1:    unchanged        +1/-0 setup.py

*** Commands ***
  1: status       2: update       3: revert       4: add untracked
  5: patch        6: diff         7: quit         8: help
What now> p
           staged     unstaged path
  1:    unchanged        +1/-0 setup.py
Patch update>> 1
           staged     unstaged path
* 1:    unchanged        +1/-0 setup.py
Patch update>>
diff --git a/setup.py b/setup.py
index fb42749..8f9d283 100755
--- a/setup.py
+++ b/setup.py
@@ -34,6 +34,7 @@ MAINTAINER_EMAIL = 'amueller@ais.uni-bonn.de'
 URL = 'http://scikit-learn.org'
 LICENSE = 'new BSD'
 DOWNLOAD_URL = 'http://sourceforge.net/projects/scikit-learn/files/'
+ADDED_JUST_FOR_YOU = "nothing"

 # We can actually import a restricted version of sklearn that
 # does not need the compiled code
Stage this hunk [y,n,q,a,d,/,e,?]? y

*** Commands ***
  1: status       2: update       3: revert       4: add untracked
  5: patch        6: diff         7: quit         8: help
What now>
```

Git gui

Repository Edit Branch Commit Merge Remote Tools Help

Current Branch: master

Unstaged Changes

- .ipynb_checkpoints/Untitled-checkpoint.ipynb
- Untitled.ipynb
- benchmarks/bench_glm.py.callgrin
- doc/.ipynb_checkpoints/Untitled-checkpoint.ipynb
- doc/Untitled.ipynb
- python35.txt
- python35_env.txt

Staged Changes (Will Commit)

- setup.py

Staged for commit File: setup.py

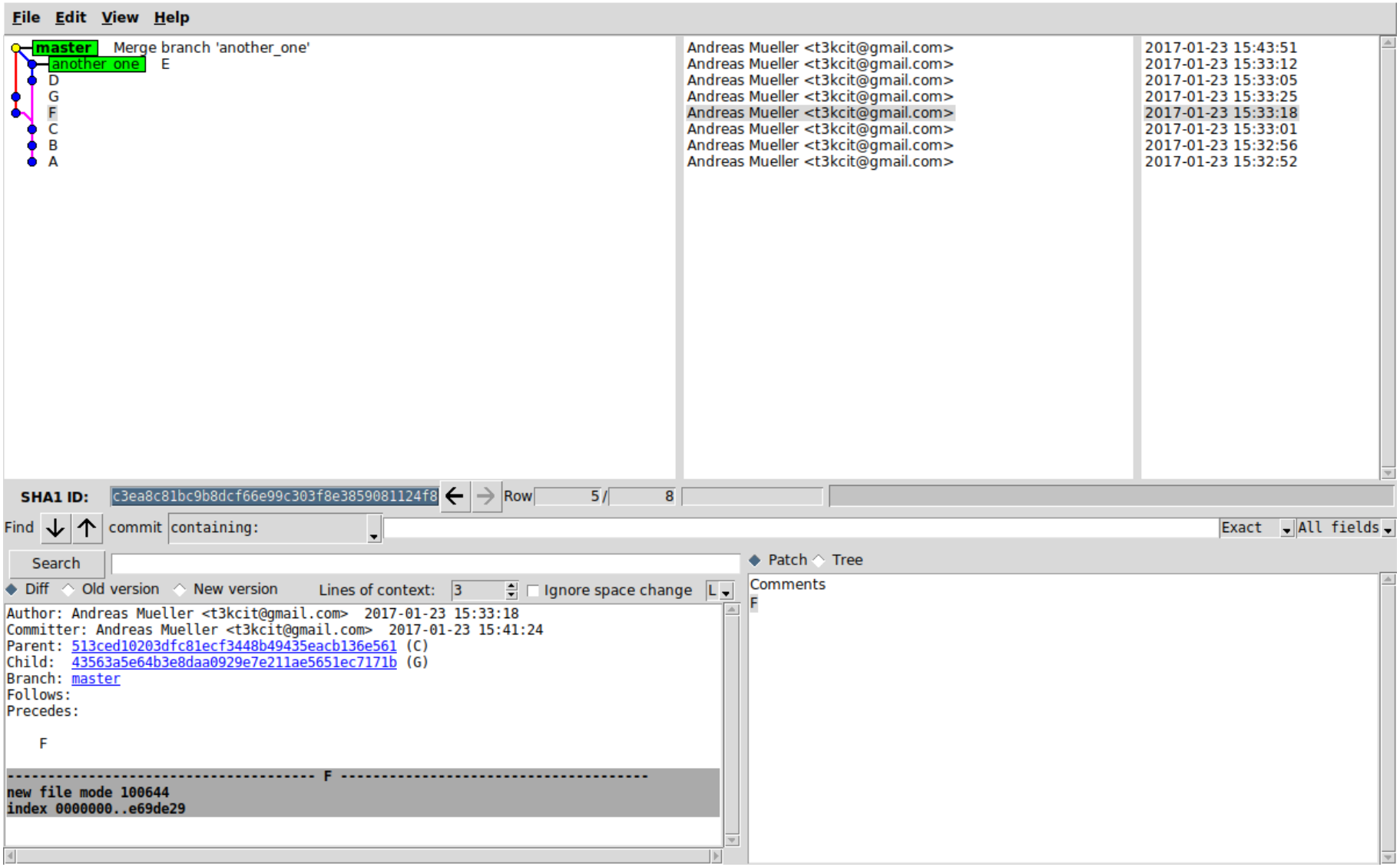
```
@@ -32,10 +32,11 @@ with open('README.rst') as f:
 MAINTAINER = 'Andreas Mueller'
 MAINTAINER_EMAIL = 'amueller@ais.uni-bonn.de'
 URL = 'http://scikit-learn.org'
 LICENSE = 'new BSD'
 DOWNLOAD_URL = 'http://sourceforge.net/projects/scikit-learn/files/'
+ADDED_JUST_FOR_YOU = "nothing"

 # We can actually import a restricted version of sklearn that
 # does not need the compiled code
 import sklearn
```

Commit Message: New Commit Amend Last Commit

Rescan
Stage Changed
Sign Off
Commit
Push

Ready.

gitk


File Edit View Help
master
Merge branch 'another_one'
another_one
E
D
G
F
C
B
A
Andreas Mueller <t3kcit@gmail.com>
2017-01-23 15:43:51
2017-01-23 15:33:12
2017-01-23 15:33:05
2017-01-23 15:33:25
2017-01-23 15:33:18
2017-01-23 15:33:01
2017-01-23 15:32:56
2017-01-23 15:32:52
SHA1 ID: c3ea8c81bc9b8dcf66e99c303f8e3859081124f8
Row 5 / 8
Find commit containing:
Exact
All fields
Search
Patch
Tree
Diff
Old version
New version
Lines of context: 3
Ignore space change
Author: Andreas Mueller <t3kcit@gmail.com> 2017-01-23 15:33:18
Committer: Andreas Mueller <t3kcit@gmail.com> 2017-01-23 15:41:24
Parent: 513ced10203dfc81ecf3448b49435eacb136e561 (C)
Child: 43563a5e64b3e8daa0929e7e211ae5651ec7171b (G)
Branch: master
Follows:
Precedes:
F
---------- F ----------
new file mode 100644
index 0000000..e69de29
Comments
F

reflog

```
git reflog
```

```
d6fedb0 HEAD@{0}: merge another_one: Merge made by the 'recursive' strategy.
43563a5 HEAD@{1}: rebase finished: returning to refs/heads/master
43563a5 HEAD@{2}: rebase: G
c3ea8c8 HEAD@{3}: rebase: F
513ced1 HEAD@{4}: rebase: checkout 513ced1
4db426c HEAD@{5}: merge feature: Fast-forward
b5ba00a HEAD@{6}: checkout: moving from master to master
b5ba00a HEAD@{7}: reset: moving to HEAD~3
6cec4ed HEAD@{8}: merge another_one: Fast-forward
513ced1 HEAD@{9}: checkout: moving from another_one to master
6cec4ed HEAD@{10}: reset: moving to 6cec4ed
4db426c HEAD@{11}: checkout: moving from feature to another_one
4db426c HEAD@{12}: checkout: moving from master to feature
513ced1 HEAD@{13}: reset: moving to HEAD~4
4db426c HEAD@{14}: checkout: moving from feature to master
4db426c HEAD@{15}: checkout: moving from master to feature
4db426c HEAD@{16}: commit: G
125e957 HEAD@{17}: commit: F
6cec4ed HEAD@{18}: commit: E
1e96a3b HEAD@{19}: commit: D
513ced1 HEAD@{20}: commit: C
b5ba00a HEAD@{21}: commit: B
db928a0 HEAD@{22}: commit (initial): A
```

```
> git log --oneline --graph --decorate --all
*   d6fedb0 (HEAD -> master) Merge branch 'another_one'
|\
| * 6cec4ed (another_one) E
| * 1e96a3b D
* | 43563a5 G
* | c3ea8c8 F
|/
* 513ced1 C
* b5ba00a B
* db928a0 A
```

Git for ages 4 and up:
https://www.youtube.com/watch?v=1ffBJ4sVUb4
(with play-doh!)

Github – just another remote!

GitHub pull request workflow

GitHub pull request workflow

End version control – but github will come back later ;)

General coding guidelines

Programs must be written for people to read, and only incidentally for machines to execute.
- Harold Abelson (wizard book)

Everyone knows that debugging is twice as hard as writing a program in the first place. So if you're as clever as you can be when you write it, how will you ever debug it?

- Brian Kernighan

Don’t be clever!
Make it readable!

Future you is the most likely person to try to understand your code.

Avoid writing code.

Python basics

Why Python

- General purpose language
- Great libraries
- Easy to learn / use
- Contenders: R (Scala?)

The two language problem

Python is sloooow...

- Numpy: C
- Scipy: C, fortran
- Pandas: Cython, Python
- Scikit-learn: Cython, Python

- CPython: C

Python 2 vs Python 3

- “current” : 2.7, 3.4, 3.5, 3.6

  Changes:
- Print
- Division
- Iterators (range, zip, map, filter, dictionary keys, values, items)
- Strings

Python 2 vs Python 3

- “current” : 2.7, 3.4, 3.5, 3.6

  Changes:
- Print
- Division
- Iterators (range, zip, map, filter, dictionary keys, values, items)
- Strings

Python 2 && Python 3

- from __future__ import print_function
- Six – tools for making 2 and 3 compatible code
- 2to3 – convert python2 code to python3 code

Python ...

Package management:

- Virtual environments
- pip (and wheels)
- Conda (and conda forge)

Python ...

Package management:

- Virtual environments
- pip (and wheels)
- Conda (and anaconda and conda forge)

Pip and upgrades

Pip upgrade works on dependencies (unless you do –no-dep)

Dynamically typed, interpreted

- Invalid syntax lying around
- Code is less self-documenting

Editors

- Flake8 / pyflake
- Scripted / weak typing: Have a syntax checker!
- use autopep8 if you have code lying around

Unit Tests and integration tests

Why test?

- Ensure that code works correctly.
- Ensure that changes don't break anything.
- Ensure that bugs are not reintroduced.
- Ensure robustness to user errors.
- Ensure code is reachable.

Test-driven development?

Types of tests

- Unit tests – function does the right thing.
- Integration tests – system / process does the right thing.
- Non-regression tests – bug got removed (and will not be reintroduced).

How to test?

- py.test – http://doc.pytest.org
- Searches for all test_*.py files, runs all test_* methods.
- Reports nice errors!

- Dig deeper: http://pybites.blogspot.com/2011/07/behind-scenes-of-pytests-new-assertion.html

Example

```python
# content of test_sample.py
def inc(x):
    return x + 2


def test_answer():
    assert inc(3) == 4
```

```
> py.test test_sample.py
============================= test session starts ==============================
platform linux -- Python 3.5.2, pytest-2.9.2, py-1.4.31, pluggy-0.3.1
rootdir: /tmp, inifile:
plugins: cov-2.4.0, timeout-1.2.0
collected 1 items

test_sample.py F

=================================== FAILURES ===================================
_________________________________ test_answer __________________________________

    def test_answer():
>       assert inc(3) == 4
E       assert 5 == 4
E        +  where 5 = inc(3)

test_sample.py:7: AssertionError
========================== 1 failed in 0.01 seconds ===========================
```

Example

```python
# content of test_sample.py
def inc(x):
    return x + 1


def test_answer():
    assert inc(3) == 4
```

```
> py.test test_sample.py
============================= test session starts ==============================
platform linux -- Python 3.5.2, pytest-2.9.2, py-1.4.31, pluggy-0.3.1
rootdir: /tmp, inifile:
plugins: cov-2.4.0, timeout-1.2.0
collected 1 items

test_sample.py .

========================== 1 passed in 0.00 seconds ==========================
```

Check coverage!

```python
# inc.py
def inc(x):
    if x < 0:
        return 0
    return x + 1


def dec(x):
    return x - 1
```

```python
# test_inc.py
from inc import inc


def test_inc():
    assert inc(3) == 4
```

```
> py.test
============================= test session starts ==============================
platform linux -- Python 3.5.2, pytest-2.9.2, py-1.4.31, pluggy-0.3.1
rootdir: /tmp/myproj, inifile:
plugins: cov-2.4.0, timeout-1.2.0
collected 1 items

test_inc.py .

========================== 1 passed in 0.01 seconds ==========================
```

Check coverage!

```python
# inc.py
def inc(x):
    if x < 0:
        return 0
    return x + 1


def dec(x):
    return x - 1
```

```python
# test_inc.py
from inc import inc


def test_inc():
    assert inc(3) == 4
```

```
/tmp/myproj [andy@dsi-amueller] [10:51]
> py.test --cov inc
============================= test session starts ==============================
platform linux -- Python 3.5.2, pytest-2.9.2, py-1.4.31, pluggy-0.3.1
rootdir: /tmp/myproj, inifile:
plugins: cov-2.4.0, timeout-1.2.0
collected 1 items

test_inc.py .

----------- coverage: platform linux, python 3.5.2-final-0 -----------
Name     Stmts   Miss  Cover
----------------------------
inc.py       6      2    67%


========================== 1 passed in 0.01 seconds ==========================
```

HTML report

```
> py.test --cov inc --cov-report=html
```

Coverage report: 67%

Module ↓	statements	missing	excluded	coverage
inc.py	6	2	0	67%
Total	**6**	**2**	**0**	**67%**

coverage.py v4.2, created at 2017-01-23 10:53

Coverage for **inc.py** : 67%

6 statements | 4 run | 2 missing | 0 excluded

```
1  # inc.py
2  def inc(x):
3      if x < 0:
4          return 0
5      return x + 1
6
7
8  def dec(x):
9      return x - 1
```

Continuous integration (with GitHub)

What is Continuous integration?

- Run command on each commit (or each PR).
- Unit testing and integration testing.
- Can act as a build-farm (for binaries or documentation).
- requires clear declaration of dependencies.
- Build matrix: Can run on many environments.
- Standard serviced: TravisCI, Jenkins and CircleCI

Benefits of CI

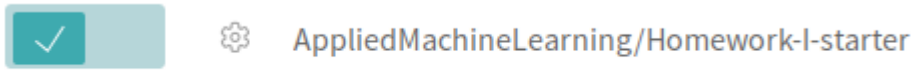
- Can run on many systems
- Can't forget to run it
- Contributor doesn't need to know details
- Can enforce style
- Can provide immediate feedback
- Protects the master branch (if run on PR)

What does it do?

- Triggered at each commit / push
- Sets up a virtual machine with your configuration.
- Pulls the current branch.
- Runs command. Usually: install, then test.
- Reports success / Failure to github.

Setting up TravisCI

- Create account linked to your github account: https://travis-ci.org/



- Check out https://docs.travis-ci.com/

```
# .travis.yml
language: python
python:
  - "2.7"
  - "3.4"
  - "3.5"
# command to install dependencies
install: "pip install -r requirements.txt"
# command to run tests
script: pytest
~
```

Using Travis

- Triggered any time you push a change
- Integrated with Pull requests
- Try a pull request on your own repository!

Documentation

Why document?

- Your code is harder to understand than you think.
- Input types and output types are unclear in dynamic languages.
- Often implicit assumptions about input.

Python documentation standards

- PEP 257 for docstrings for class, methods and functions

```
def inc(x):
    """Add one to a number.

    This function takes as argument a number,
    and adds one to it.
    The result is returned.
    """
    if x < 0:
        return 0
    return x + 1
```

- Additional inline documentation

```
if x < 0:
    # x is less then zero
    return 0
return x + 1
```

NumpyDoc format

- See https://github.com/numpy/numpy/blob/master/doc/HOWTO_DOCUMENT.rst.txt

```
Parameters
----------
x : type
    Description of parameter `x`.
```

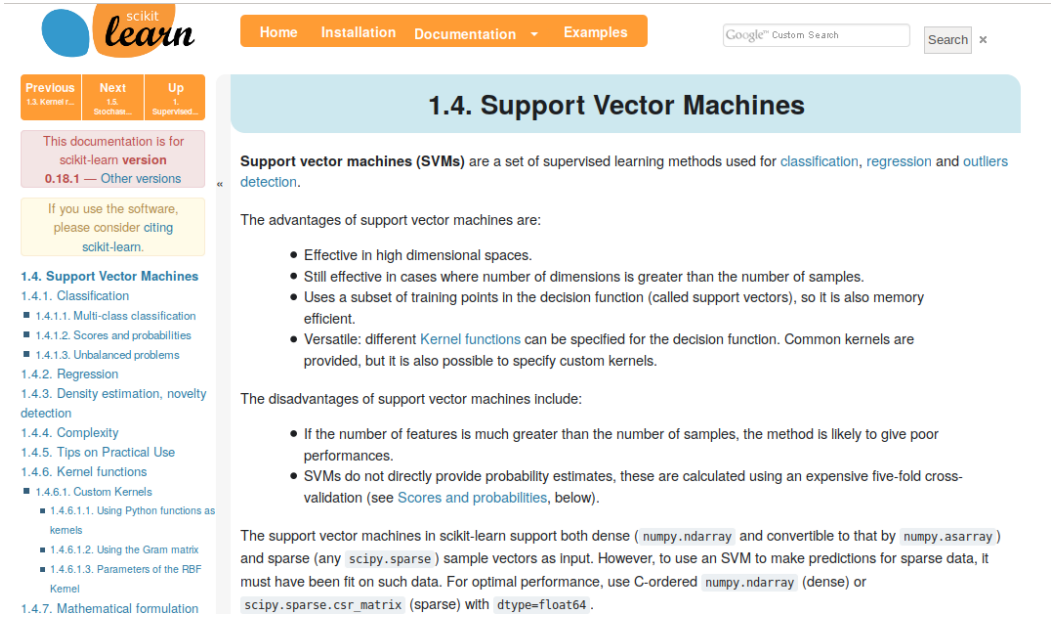
Examples

```python
class MultinomialNB(BaseDiscreteNB):
    """
    Naive Bayes classifier for multinomial models

    The multinomial Naive Bayes classifier is suitable for classification with
    discrete features (e.g., word counts for text classification). The
    multinomial distribution normally requires integer feature counts. However,
    in practice, fractional counts such as tf-idf may also work.

    Read more in the :ref:`User Guide <multinomial_naive_bayes>`.

    Parameters
    ----------
    alpha : float, optional (default=1.0)
        Additive (Laplace/Lidstone) smoothing parameter
        (0 for no smoothing).

    fit_prior : boolean, optional (default=True)
        Whether to learn class prior probabilities or not.
        If false, a uniform prior will be used.

    class_prior : array-like, size (n_classes,), optional (default=None)
        Prior probabilities of the classes. If specified the priors are not
        adjusted according to the data.

    Attributes
    ----------
    class_log_prior_ : array, shape (n_classes, )
        Smoothed empirical log probability for each class.

    intercept_ : property
        Mirrors ``class_log_prior_`` for interpreting MultinomialNB
        as a linear model.

    feature_log_prob_ : array, shape (n_classes, n_features)
        Empirical log probability of features
        given a class, ``P(x_i|y)``.

    coef_ : property
        Mirrors ``feature_log_prob_`` for interpreting MultinomialNB
        as a linear model.

    class_count_ : array, shape (n_classes,)
        Number of samples encountered for each class during fitting. This
        value is weighted by the sample weight when provided.

    feature_count_ : array, shape (n_classes, n_features)
        Number of samples encountered for each (class, feature)
        during fitting. This value is weighted by the sample weight when
        provided.

    Examples
    --------
    >>> import numpy as np
    >>> X = np.random.randint(5, size=(6, 100))
    >>> y = np.array([1, 2, 3, 4, 5, 6])
    >>> from sklearn.naive_bayes import MultinomialNB
    >>> clf = MultinomialNB()
    >>> clf.fit(X, y)
    MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
    >>> print(clf.predict(X[2:3]))
    [3]

    Notes
    -----
    For the rationale behind the names `coef_` and `intercept_`, i.e.
    naive Bayes as a linear classifier, see J. Rennie et al. (2003),
    Tackling the poor assumptions of naive Bayes text classifiers, ICML.

    References
    ----------
    C.D. Manning, P. Raghavan and H. Schuetze (2008). Introduction to
    Information Retrieval. Cambridge University Press, pp. 234-265.
    http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html
    """
```

```python
def fit(self, X, y, sample_weight=None):
    """Fit Naive Bayes classifier according to X, y

    Parameters
    ----------
    X : {array-like, sparse matrix}, shape = [n_samples, n_features]
        Training vectors, where n_samples is the number of samples and
        n_features is the number of features.

    y : array-like, shape = [n_samples]
        Target values.

    sample_weight : array-like, shape = [n_samples], optional (default=None)
        Weights applied to individual samples (1. for unweighted).

    Returns
    -------
    self : object
        Returns self.
    """
```

Sphinx and reStructured Text

```
.. _svm:

=======================
Support Vector Machines
=======================

.. currentmodule:: sklearn.svm

**Support vector machines (SVMs)** are a set of supervised learning
methods used for :ref:`classification <svm_classification>`,
:ref:`regression <svm_regression>` and :ref:`outliers detection
<svm_outlier_detection>`.

The advantages of support vector machines are:

    - Effective in high dimensional spaces.

    - Still effective in cases where number of dimensions is greater
      than the number of samples.

    - Uses a subset of training points in the decision function (called
      support vectors), so it is also memory efficient.

    - Versatile: different :ref:`svm_kernels` can be
      specified for the decision function. Common kernels are
      provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

    - If the number of features is much greater than the number of
      samples, the method is likely to give poor performances.

    - SVMs do not directly provide probability estimates, these are
      calculated using an expensive five-fold cross-validation
      (see :ref:`Scores and probabilities <scores_probabilities>`, below).

The support vector machines in scikit-learn support both dense
(``numpy.ndarray`` and convertible to that by ``numpy.asarray``) and
sparse (any ``scipy.sparse``) sample vectors as input. However, to use
an SVM to make predictions for sparse data, it must have been fit on such
data. For optimal performance, use C-ordered ``numpy.ndarray`` (dense) or
``scipy.sparse.csr_matrix`` (sparse) with ``dtype=float64``.
```



Autodoc and NumpyDoc

sklearn.dummy.DummyClassifier

class `sklearn.dummy.` **`DummyClassifier`** (*strategy='stratified', random_state=None, constant=None*) [source]

DummyClassifier is a classifier that makes predictions using simple rules.

This classifier is useful as a simple baseline to compare with other (real) classifiers. Do not use it for real problems.

Read more in the User Guide.

Parameters:

strategy : str, default="stratified"

Strategy to use to generate predictions.

- "stratified": generates predictions by respecting the training set's class distribution.
- "most_frequent": always predicts the most frequent label in the training set.
- "prior": always predicts the class that maximizes the class prior (like "most_frequent") and `predict_proba` returns the class prior.
- "uniform": generates predictions uniformly at random.
- "constant": always predicts a constant label that is provided by the user. This is useful for metrics that evaluate a non-majority class

New in version 0.17: Dummy Classifier now supports prior fitting strategy using parameter *prior*.

random_state : int seed, RandomState instance, or None (default)

The seed of the pseudo random number generator to use.

constant : int or str or array of shape = [n_outputs]

The explicit constant as predicted by the "constant" strategy. This parameter is useful only for the "constant" strategy.

Attributes:

classes_ : array or list of array of shape = [n_classes]

Class labels for each output.

n_classes_ : array or list of array of shape = [n_classes]

Setting up Sphinx

- Install sphinx
- Run sphinx-autogen
- Edit conf.py – pick a theme like https://github.com/snide/sphinx_rtd_theme
- Edit your index.rst

Setting up ReadTheDocs