

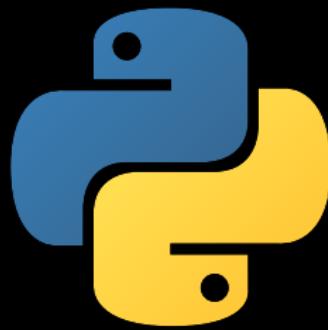
2 The tool: a Python library



©Theodore W. Gray

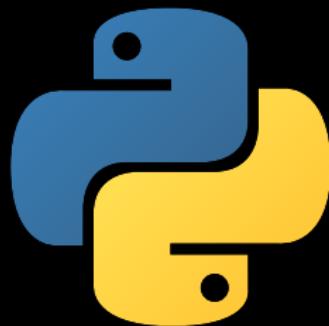
2 Python for scientific computing

*Scientific computing
needs a high-level
interactive
environment*



2 Python for scientific computing

*Scientific computing
needs a high-level
interactive
environment*



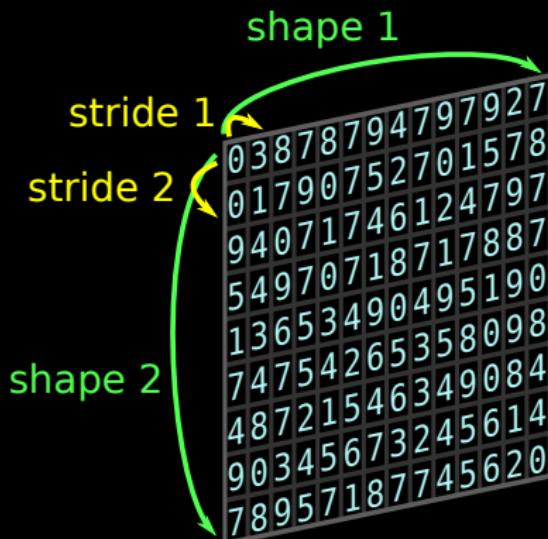
Memory management
⇒ a virtual machine

Python's virtual machine is rudimentary
Enables low-level computation
and coupling to numerical libraries

2 numpy: array computing in Python

numpy =

- memory & data specification
- reshaping with minimal copies
- semantics of operations



Represents any regular data in a structured way

2 numpy: array computing in Python

numpy =

- memory & data specification
- reshaping with minimal copies
- semantics of operations

Matches the memory model of numerical libraries

⇒ Enables copyless interactions

Numpy is really a memory model

shape 2



1	3	6	5	3	4	9	0
7	4	7	5	4	2	6	5
3	5	8	0	9	8	0	8
4	8	7	2	1	5	4	6
6	3	4	5	6	7	3	2
9	0	3	4	5	6	7	4
5	7	8	9	5	7	1	8
2	0	6	2	0	5	6	2

2 The scientific-computing ecosystem



- **scipy**: numerical algorithms
 - linear algebra
 - optimization
 - special function
 - interpolation
 - statistical functions
 - FFT
 - **matplotlib**: plotting
matlab-like plotting and beyond
 - **pandas**: columnar data
programmatic excel
 - **scikit-image**: image processing
 - **statsmodels**: statistical models
- ...

2 The scientific-computing ecosystem

- **scipy**: numerical algorithms

- linear algebra
- optimization
- special function
- interpolation
- statistical functions
- FFT



- **matplotlib**: plotting

<http://scipy-lectures.org>

- **pandas**: columnar data

programmatic excel

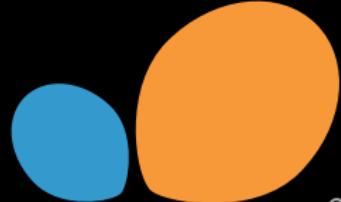
- **scikit-image**: image processing

- **statsmodels**: statistical models

...

A library, not a program

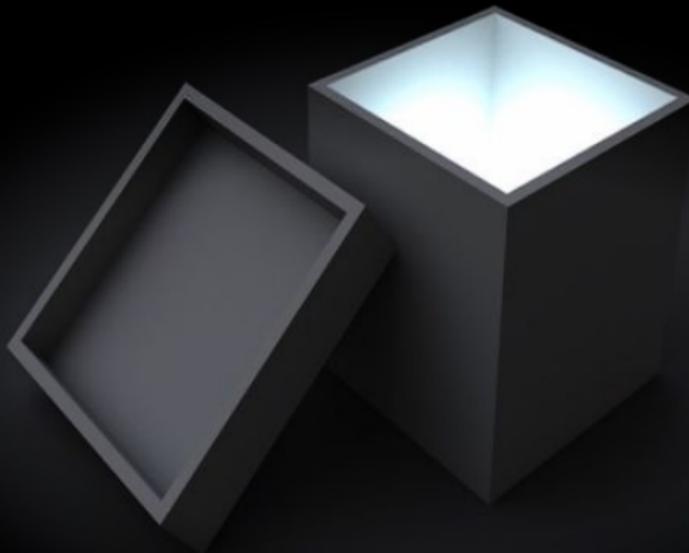
- More expressive and flexible
- Easy to include in an ecosystem



The greybox model

Building bricks

to combine with domain-specific knowledge
interchangeable (mostly)



The greybox model

```
from sklearn import svm  
classifier = svm.SVC()  
classifier.fit(X_train, Y_train)  
Y_test = classifier.predict(X_test)
```

Access to the model's inner parameters

```
coef = classifier.coef_
```

From raw data to a sample matrix X

- For text data: counting word occurrences
 - Input data: list of documents (string)
 - Output data: numerical matrix



data Python
more implementation
supervised efficient While
efficient some BSD statistical problems
some method implementations Computing
method shogun pypy large
shogun other libsvm input objects compiled
other estimators platforms need use license
estimators input numpy & array pypy optimizations provides
input object used scikit-learn interface
used provide set linear PCA libraries code memory
provide score engineering estimator high-level parameters
score engineering estimator Journal code documentation
engineering estimator algorithms example MDP analysis performance
engineering estimator machine scientific language
Scikit-learn learning

From raw data to a sample matrix \mathbf{X}

- For text data: counting word occurrences
 - Input data: list of documents (string)
 - Output data: numerical matrix

```
from sklearn.feature_extraction.text  
      import HashingVectorizer  
hasher = HashingVectorizer()  
  
X = hasher.fit_transform(documents)
```

Supervised learning

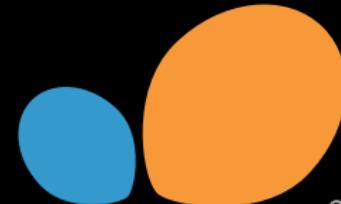
- Decision trees (Random-Forest, Boosted Tree)
- Linear models ■ SVM
- Gaussian processes ...

Unsupervised Learning

- Clustering ■ Mixture models
- Dictionary learning ■ ICA
- Outlier detection ...

Model selection

- Cross-validation
- Parameter optimization



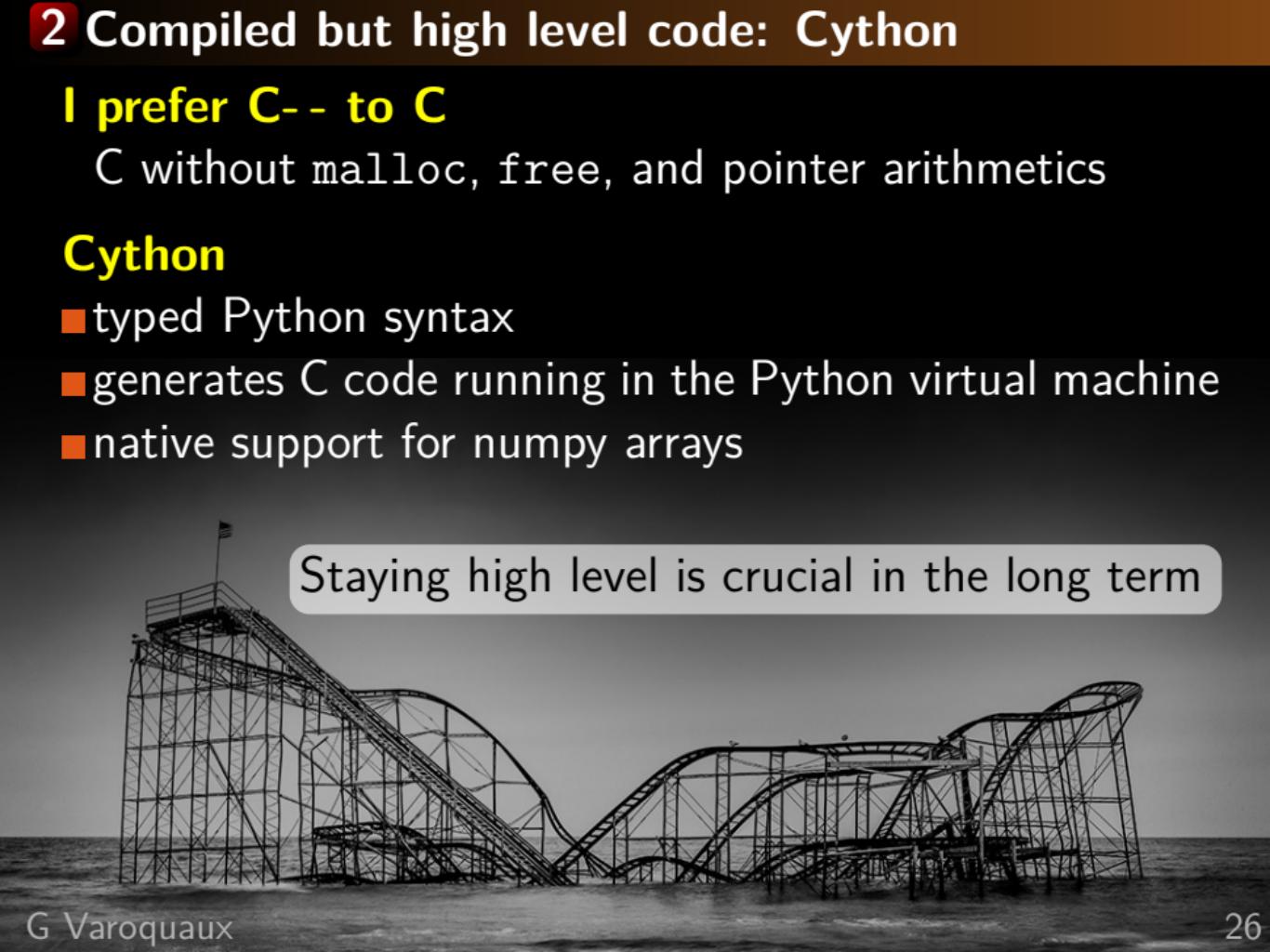
2 Compiled but high level code: Cython

I prefer C- to C

C without malloc, free, and pointer arithmetics

Cython

- typed Python syntax
- generates C code running in the Python virtual machine
- native support for numpy arrays



Staying high level is crucial in the long term

Some gems in scikit-learn

SAG:

```
linear_model.LogisticRegression(solver='sag')
```

Fast linear model on biggish data



Some gems in scikit-learn

SAG:

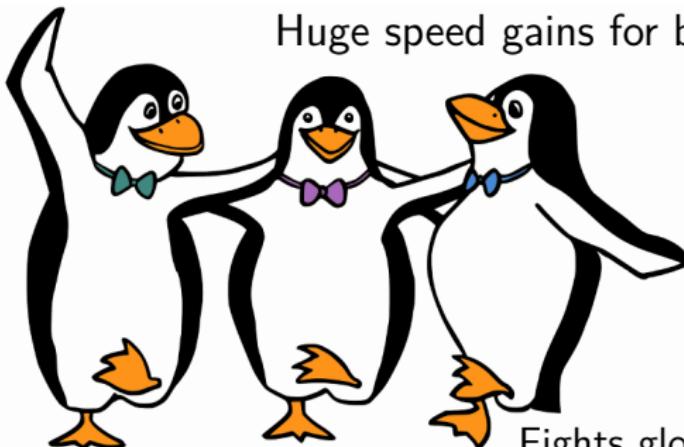
```
linear_model.LogisticRegression(solver='sag')
```

Fast linear model on biggish data

PCA == RandomizedPCA: (0.18)

Heuristic to switch PCA to random linear algebra

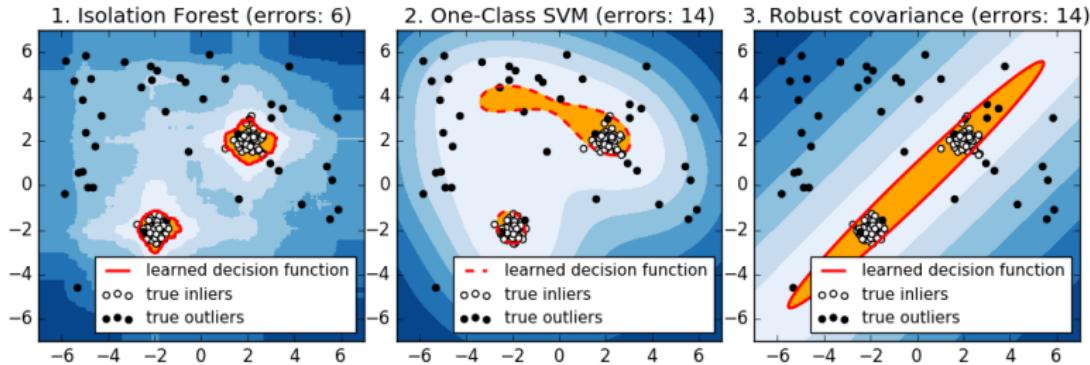
Huge speed gains for biggish data



Fights global warming

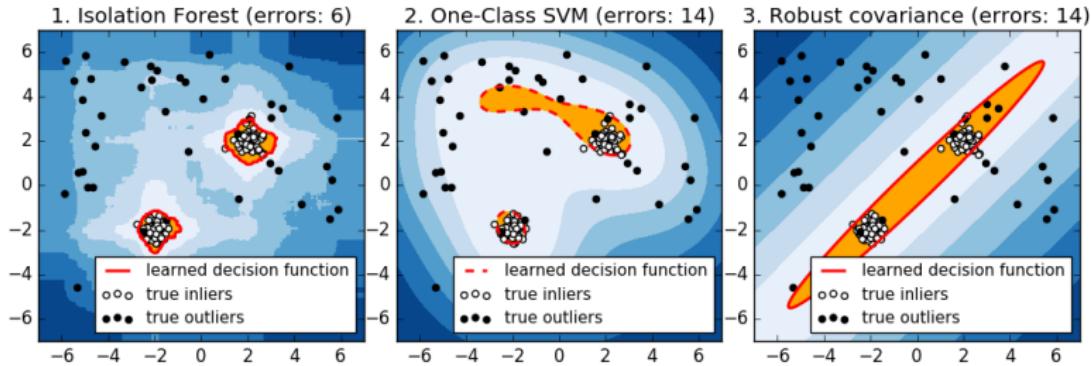
Some gems in scikit-learn

Outlier detection and isolation forests (0.18)



Some gems in scikit-learn

Outlier detection and isolation forests (0.18)



Distributed computing (soon)

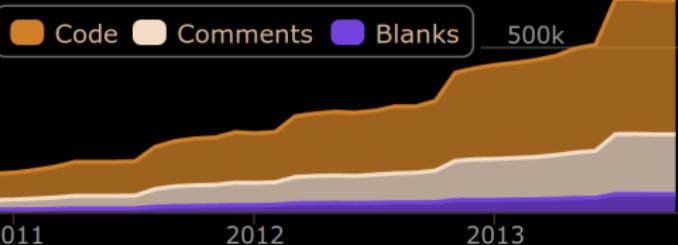
dask & hadoop backends to joblib

3 The project: a community



3 Community-based development in scikit-learn

Huge feature set:
benefits of a large team



Project growth:

- More than 200 contributors
- ~ 12 core contributors
- 1 full-time INRIA programmer from the start



Estimated cost of development: \$ 6 millions

COCOMO model,

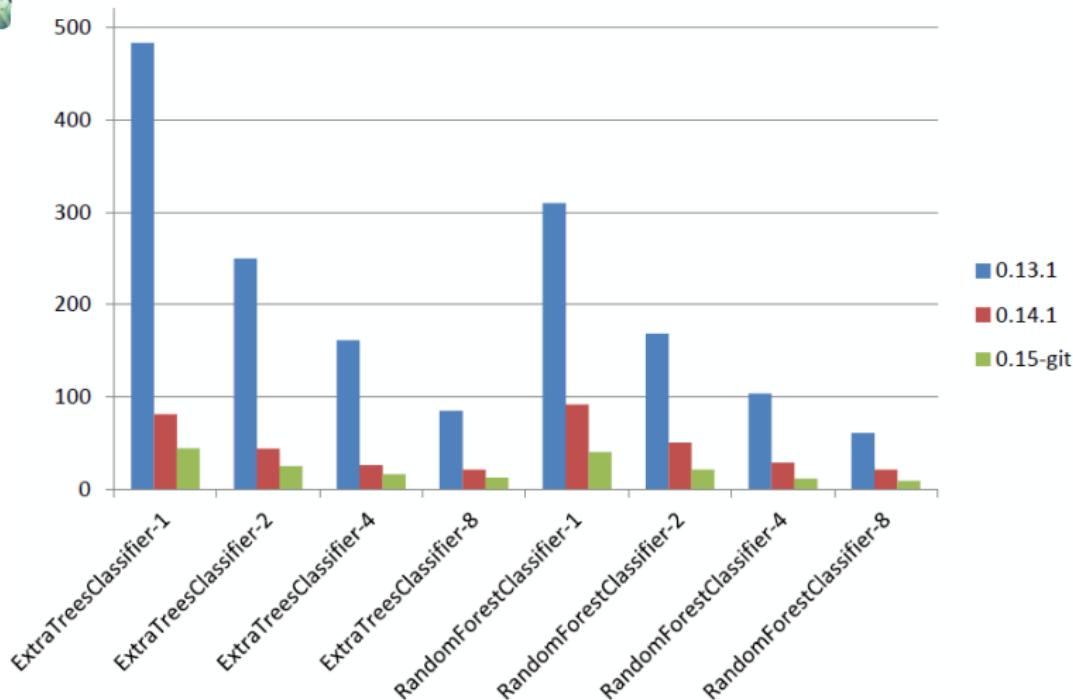
<http://www.ohloh.net/p/scikit-learn>

3 Many eyes makes code fast



Gilles Louuppe @glouppe · Feb 18

Speed improvement from 0.13 to 0.15-git of Random Forests in Scikit-Learn:



L. Buitinck, O. Grisel, A. Joly, G. Louuppe, J. Nothman, P. Prettenhofer

3 6 steps to a community-driven project

- 1 Focus on **quality**
- 2 Build great **docs and examples**
- 3 Use **github**
- 4 Limit the technicality of your codebase
- 5 Releasing and packaging matter
- 6 Focus on your contributors,
give them credit, decision power



[http://www.slideshare.net/GaelVaroquaux/
scikit-learn-dvelopement-communautaire](http://www.slideshare.net/GaelVaroquaux/scikit-learn-dvelopement-communautaire)

G Varoquaux

3 Quality assurance

Unit testing

- Everything is tested
If it's not tested, it's broken
- Test API
Test as grey box
- Test numerics
Check mathematical properties
(eg decrease of energy)
- Tests should run fast
- Perfect control of randomness



3 Quality assurance

Code review: pull requests

- Can include newcomers
- We read each others code
- Everything is discussed:
 - Should the algorithm go in?
 - Are there good defaults?
 - Are names meaningful?
 - Are the numerics stable?
 - Could it be faster?

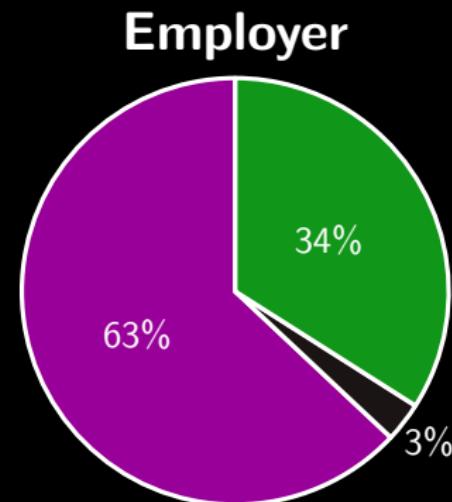
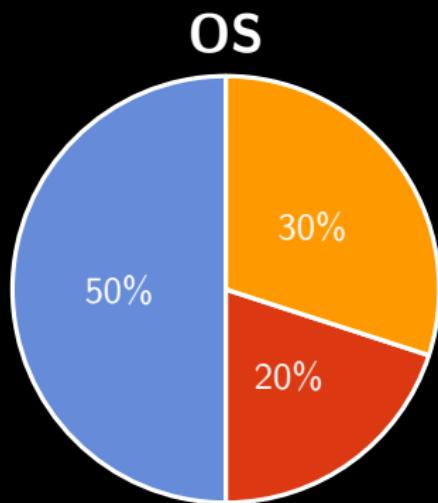
The screenshot shows a GitHub pull request interface. At the top, a message from 'agramfort' states: 'agramfort started a discussion in the diff 7 months ago'. Below this is a code snippet from 'sklearn/cluster/_inertia.pyx'. Lines 21 through 39 are shown, with line 24 highlighted in red and line 39 highlighted in green. A comment from 'agramfort' below the code asks: 'i am afraid this is numerically less stable. It is justified by speed?'. A reply from 'jmetzen' follows: 'you are right, I reverted it to the old implementation'. At the bottom, a commit message from 'jmetzen' is shown: '8e00db8 ENH: CompleteLinkage supports more efficient'.

```
...  ... @@ -21,9 +36,9 @@ def compute_ward_dist(np.ndarray[D
21   36         for i in range(size_max):
22   37             row = coord_row[i]
23   38             col = coord_col[i]
24 -             n = (m_1[row] * m_1[col]) / (m_1[row] + m_1
25 +             n = 1.0 / (1.0 / m_1[row] + 1.0 / m_1[col])
26
27             if n <= 0.0:
28                 continue
29
30             if n >= 1.0:
31                 return 0.0
32
33             if n <= 0.5:
34                 continue
35
36             if n >= 0.75:
37                 return 0.0
38
39             if n >= 0.5:
40                 continue
41
42             if n >= 0.75:
43                 return 0.0
44
45             if n >= 0.5:
46                 continue
47
48             if n >= 0.75:
49                 return 0.0
50
51             if n >= 0.5:
52                 continue
53
54             if n >= 0.75:
55                 return 0.0
56
57             if n >= 0.5:
58                 continue
59
5
60             if n >= 0.75:
61                 return 0.0
62
63             if n >= 0.5:
64                 continue
65
66             if n >= 0.75:
67                 return 0.0
68
69             if n >= 0.5:
70                 continue
71
72             if n >= 0.75:
73                 return 0.0
74
75             if n >= 0.5:
76                 continue
77
78             if n >= 0.75:
79                 return 0.0
80
81             if n >= 0.5:
82                 continue
83
84             if n >= 0.75:
85                 return 0.0
86
87             if n >= 0.5:
88                 continue
89
90             if n >= 0.75:
91                 return 0.0
92
93             if n >= 0.5:
94                 continue
95
96             if n >= 0.75:
97                 return 0.0
98
99             if n >= 0.5:
100                continue
101
102             if n >= 0.75:
103                 return 0.0
104
105             if n >= 0.5:
106                 continue
107
108             if n >= 0.75:
109                 return 0.0
110
111             if n >= 0.5:
112                 continue
113
114             if n >= 0.75:
115                 return 0.0
116
117             if n >= 0.5:
118                 continue
119
11
120             if n >= 0.75:
121                 return 0.0
122
123             if n >= 0.5:
124                 continue
125
126             if n >= 0.75:
127                 return 0.0
128
129             if n >= 0.5:
130                 continue
131
132             if n >= 0.75:
133                 return 0.0
134
135             if n >= 0.5:
136                 continue
137
138             if n >= 0.75:
139                 return 0.0
140
141             if n >= 0.5:
142                 continue
143
144             if n >= 0.75:
145                 return 0.0
146
147             if n >= 0.5:
148                 continue
149
14
150             if n >= 0.75:
151                 return 0.0
152
153             if n >= 0.5:
154                 continue
155
156             if n >= 0.75:
157                 return 0.0
158
159             if n >= 0.5:
160                 continue
161
162             if n >= 0.75:
163                 return 0.0
164
165             if n >= 0.5:
166                 continue
167
168             if n >= 0.75:
169                 return 0.0
170
171             if n >= 0.5:
172                 continue
173
174             if n >= 0.75:
175                 return 0.0
176
177             if n >= 0.5:
178                 continue
179
180             if n >= 0.75:
181                 return 0.0
182
183             if n >= 0.5:
184                 continue
185
186             if n >= 0.75:
187                 return 0.0
188
189             if n >= 0.5:
190                 continue
191
192             if n >= 0.75:
193                 return 0.0
194
195             if n >= 0.5:
196                 continue
197
198             if n >= 0.75:
199                 return 0.0
19
200             if n >= 0.5:
201                 continue
202
203             if n >= 0.75:
204                 return 0.0
205
206             if n >= 0.5:
207                 continue
208
209             if n >= 0.75:
210                 return 0.0
211
212             if n >= 0.5:
213                 continue
214
215             if n >= 0.75:
216                 return 0.0
217
218             if n >= 0.5:
219                 continue
220
221             if n >= 0.75:
222                 return 0.0
223
224             if n >= 0.5:
225                 continue
226
227             if n >= 0.75:
228                 return 0.0
229
230             if n >= 0.5:
231                 continue
232
233             if n >= 0.75:
234                 return 0.0
235
236             if n >= 0.5:
237                 continue
238
239             if n >= 0.75:
240                 return 0.0
241
242             if n >= 0.5:
243                 continue
244
245             if n >= 0.75:
246                 return 0.0
247
248             if n >= 0.5:
249                 continue
250
251             if n >= 0.75:
252                 return 0.0
253
254             if n >= 0.5:
255                 continue
256
257             if n >= 0.75:
258                 return 0.0
259
260             if n >= 0.5:
261                 continue
262
263             if n >= 0.75:
264                 return 0.0
265
266             if n >= 0.5:
267                 continue
268
269             if n >= 0.75:
270                 return 0.0
271
272             if n >= 0.5:
273                 continue
274
275             if n >= 0.75:
276                 return 0.0
277
278             if n >= 0.5:
279                 continue
280
281             if n >= 0.75:
282                 return 0.0
283
284             if n >= 0.5:
285                 continue
286
287             if n >= 0.75:
288                 return 0.0
289
290             if n >= 0.5:
291                 continue
292
293             if n >= 0.75:
294                 return 0.0
295
296             if n >= 0.5:
297                 continue
298
299             if n >= 0.75:
29
300             if n >= 0.5:
301                 continue
302
303             if n >= 0.75:
304                 return 0.0
305
306             if n >= 0.5:
307                 continue
308
309             if n >= 0.75:
310                 return 0.0
311
312             if n >= 0.5:
313                 continue
314
315             if n >= 0.75:
316                 return 0.0
317
318             if n >= 0.5:
319                 continue
320
321             if n >= 0.75:
322                 return 0.0
323
324             if n >= 0.5:
325                 continue
326
327             if n >= 0.75:
328                 return 0.0
329
330             if n >= 0.5:
331                 continue
332
333             if n >= 0.75:
334                 return 0.0
335
336             if n >= 0.5:
337                 continue
338
339             if n >= 0.75:
340                 return 0.0
341
342             if n >= 0.5:
343                 continue
344
345             if n >= 0.75:
346                 return 0.0
347
348             if n >= 0.5:
349                 continue
350
351             if n >= 0.75:
352                 return 0.0
353
354             if n >= 0.5:
355                 continue
356
357             if n >= 0.75:
358                 return 0.0
359
360             if n >= 0.5:
361                 continue
362
363             if n >= 0.75:
364                 return 0.0
365
366             if n >= 0.5:
367                 continue
368
369             if n >= 0.75:
370                 return 0.0
371
372             if n >= 0.5:
373                 continue
374
375             if n >= 0.75:
376                 return 0.0
377
378             if n >= 0.5:
379                 continue
380
381             if n >= 0.75:
382                 return 0.0
383
384             if n >= 0.5:
385                 continue
386
387             if n >= 0.75:
388                 return 0.0
389
390             if n >= 0.5:
391                 continue
392
393             if n >= 0.75:
394                 return 0.0
395
396             if n >= 0.5:
397                 continue
398
399             if n >= 0.75:
400                 return 0.0
401
402             if n >= 0.5:
403                 continue
404
405             if n >= 0.75:
406                 return 0.0
407
408             if n >= 0.5:
409                 continue
410
411             if n >= 0.75:
412                 return 0.0
413
414             if n >= 0.5:
415                 continue
416
417             if n >= 0.75:
418                 return 0.0
419
420             if n >= 0.5:
421                 continue
422
423             if n >= 0.75:
424                 return 0.0
425
426             if n >= 0.5:
427                 continue
428
429             if n >= 0.75:
430                 return 0.0
431
432             if n >= 0.5:
433                 continue
434
435             if n >= 0.75:
436                 return 0.0
437
438             if n >= 0.5:
439                 continue
440
441             if n >= 0.75:
442                 return 0.0
443
444             if n >= 0.5:
445                 continue
446
447             if n >= 0.75:
448                 return 0.0
449
450             if n >= 0.5:
451                 continue
452
453             if n >= 0.75:
454                 return 0.0
455
456             if n >= 0.5:
457                 continue
458
459             if n >= 0.75:
460                 return 0.0
461
462             if n >= 0.5:
463                 continue
464
465             if n >= 0.75:
466                 return 0.0
467
468             if n >= 0.5:
469                 continue
470
471             if n >= 0.75:
472                 return 0.0
473
474             if n >= 0.5:
475                 continue
476
477             if n >= 0.75:
478                 return 0.0
479
480             if n >= 0.5:
481                 continue
482
483             if n >= 0.75:
484                 return 0.0
485
486             if n >= 0.5:
487                 continue
488
489             if n >= 0.75:
490                 return 0.0
491
492             if n >= 0.5:
493                 continue
494
495             if n >= 0.75:
496                 return 0.0
497
498             if n >= 0.5:
499                 continue
500
501             if n >= 0.75:
502                 return 0.0
503
504             if n >= 0.5:
505                 continue
506
507             if n >= 0.75:
508                 return 0.0
509
510             if n >= 0.5:
511                 continue
512
513             if n >= 0.75:
514                 return 0.0
515
516             if n >= 0.5:
517                 continue
518
519             if n >= 0.75:
520                 return 0.0
521
522             if n >= 0.5:
523                 continue
524
525             if n >= 0.75:
526                 return 0.0
527
528             if n >= 0.5:
529                 continue
530
531             if n >= 0.75:
532                 return 0.0
533
534             if n >= 0.5:
535                 continue
536
537             if n >= 0.75:
538                 return 0.0
539
540             if n >= 0.5:
541                 continue
542
543             if n >= 0.75:
544                 return 0.0
545
546             if n >= 0.5:
547                 continue
548
549             if n >= 0.75:
550                 return 0.0
551
552             if n >= 0.5:
553                 continue
554
555             if n >= 0.75:
556                 return 0.0
557
558             if n >= 0.5:
559                 continue
560
561             if n >= 0.75:
562                 return 0.0
563
564             if n >= 0.5:
565                 continue
566
567             if n >= 0.75:
568                 return 0.0
569
570             if n >= 0.5:
571                 continue
572
573             if n >= 0.75:
574                 return 0.0
575
576             if n >= 0.5:
577                 continue
578
579             if n >= 0.75:
580                 return 0.0
581
582             if n >= 0.5:
583                 continue
584
585             if n >= 0.75:
586                 return 0.0
587
588             if n >= 0.5:
589                 continue
590
591             if n >= 0.75:
592                 return 0.0
593
594             if n >= 0.5:
595                 continue
596
597             if n >= 0.75:
598                 return 0.0
599
600             if n >= 0.5:
601                 continue
602
603             if n >= 0.75:
604                 return 0.0
605
606             if n >= 0.5:
607                 continue
608
609             if n >= 0.75:
610                 return 0.0
611
612             if n >= 0.5:
613                 continue
614
615             if n >= 0.75:
616                 return 0.0
617
618             if n >= 0.5:
619                 continue
620
621             if n >= 0.75:
622                 return 0.0
623
624             if n >= 0.5:
625                 continue
626
627             if n >= 0.75:
628                 return 0.0
629
630             if n >= 0.5:
631                 continue
632
633             if n >= 0.75:
634                 return 0.0
635
636             if n >= 0.5:
637                 continue
638
639             if n >= 0.75:
640                 return 0.0
641
642             if n >= 0.5:
643                 continue
644
645             if n >= 0.75:
646                 return 0.0
647
648             if n >= 0.5:
649                 continue
650
651             if n >= 0.75:
652                 return 0.0
653
654             if n >= 0.5:
655                 continue
656
657             if n >= 0.75:
658                 return 0.0
659
660             if n >= 0.5:
661                 continue
662
663             if n >= 0.75:
664                 return 0.0
665
666             if n >= 0.5:
667                 continue
668
669             if n >= 0.75:
670                 return 0.0
671
672             if n >= 0.5:
673                 continue
674
675             if n >= 0.75:
676                 return 0.0
677
678             if n >= 0.5:
679                 continue
680
681             if n >= 0.75:
682                 return 0.0
683
684             if n >= 0.5:
685                 continue
686
687             if n >= 0.75:
688                 return 0.0
689
690             if n >= 0.5:
691                 continue
692
693             if n >= 0.75:
694                 return 0.0
695
696             if n >= 0.5:
697                 continue
698
699             if n >= 0.75:
700                 return 0.0
701
702             if n >= 0.5:
703                 continue
704
705             if n >= 0.75:
706                 return 0.0
707
708             if n >= 0.5:
709                 continue
710
711             if n >= 0.75:
712                 return 0.0
713
714             if n >= 0.5:
715                 continue
716
717             if n >= 0.75:
718                 return 0.0
719
720             if n >= 0.5:
721                 continue
722
723             if n >= 0.75:
724                 return 0.0
725
726             if n >= 0.5:
727                 continue
728
729             if n >= 0.75:
730                 return 0.0
731
732             if n >= 0.5:
733                 continue
734
735             if n >= 0.75:
736                 return 0.0
737
738             if n >= 0.5:
739                 continue
740
741             if n >= 0.75:
742                 return 0.0
743
744             if n >= 0.5:
745                 continue
746
747             if n >= 0.75:
748                 return 0.0
749
750             if n >= 0.5:
751                 continue
752
753             if n >= 0.75:
754                 return 0.0
755
756             if n >= 0.5:
757                 continue
758
759             if n >= 0.75:
760                 return 0.0
761
762             if n >= 0.5:
763                 continue
764
765             if n >= 0.75:
766                 return 0.0
767
768             if n >= 0.5:
769                 continue
770
771             if n >= 0.75:
772                 return 0.0
773
774             if n >= 0.5:
775                 continue
776
777             if n >= 0.75:
778                 return 0.0
779
780             if n >= 0.5:
781                 continue
782
783             if n >= 0.75:
784                 return 0.0
785
786             if n >= 0.5:
787                 continue
788
789             if n >= 0.75:
790                 return 0.0
791
792             if n >= 0.5:
793                 continue
794
795             if n >= 0.75:
796                 return 0.0
797
798             if n >= 0.5:
799                 continue
800
801             if n >= 0.75:
802                 return 0.0
803
804             if n >= 0.5:
805                 continue
806
807             if n >= 0.75:
808                 return 0.0
809
810             if n >= 0.5:
811                 continue
812
813             if n >= 0.75:
814                 return 0.0
815
816             if n >= 0.5:
817                 continue
818
819             if n >= 0.75:
820                 return 0.0
821
822             if n >= 0.5:
823                 continue
824
825             if n >= 0.75:
826                 return 0.0
827
828             if n >= 0.5:
829                 continue
830
831             if n >= 0.75:
832                 return 0.0
833
834             if n >= 0.5:
835                 continue
836
837             if n >= 0.75:
838                 return 0.0
839
840             if n >= 0.5:
841                 continue
842
843             if n >= 0.75:
844                 return 0.0
845
846             if n >= 0.5:
847                 continue
848
849             if n >= 0.75:
850                 return 0.0
851
852             if n >= 0.5:
853                 continue
854
855             if n >= 0.75:
856                 return 0.0
857
858             if n >= 0.5:
859                 continue
860
861             if n >= 0.75:
862                 return 0.0
863
864             if n >= 0.5:
865                 continue
866
867             if n >= 0.75:
868                 return 0.0
869
870             if n >= 0.5:
871                 continue
872
873             if n >= 0.75:
874                 return 0.0
875
876             if n >= 0.5:
877                 continue
878
879             if n >= 0.75:
880                 return 0.0
881
882             if n >= 0.5:
883                 continue
884
885             if n >= 0.75:
886                 return 0.0
887
888             if n >= 0.5:
889                 continue
890
891             if n >= 0.75:
892                 return 0.0
893
894             if n >= 0.5:
895                 continue
896
897             if n >= 0.75:
898                 return 0.0
899
900             if n >= 0.5:
901                 continue
902
903             if n >= 0.75:
904                 return 0.0
905
906             if n >= 0.5:
907                 continue
908
909             if n >= 0.75:
910                 return 0.0
911
912             if n >= 0.5:
913                 continue
914
915             if n >= 0.75:
916                 return 0.0
917
918             if n >= 0.5:
919                 continue
920
921             if n >= 0.75:
922                 return 0.0
923
924             if n >= 0.5:
925                 continue
926
927             if n >= 0.75:
928                 return 0.0
929
930             if n >= 0.5:
931                 continue
932
933             if n >= 0.75:
934                 return 0.0
935
936             if n >= 0.5:
937                 continue
938
939             if n >= 0.75:
940                 return 0.0
941
942             if n >= 0.5:
943                 continue
944
945             if n >= 0.75:
946                 return 0.0
947
948             if n >= 0.5:
949                 continue
950
951             if n >= 0.75:
952                 return 0.0
953
954             if n >= 0.5:
955                 continue
956
957             if n >= 0.75:
958                 return 0.0
959
960             if n >= 0.5:
961                 continue
962
963             if n >= 0.75:
964                 return 0.0
965
966             if n >= 0.5:
967                 continue
968
969             if n >= 0.75:
970                 return 0.0
971
972             if n >= 0.5:
973                 continue
974
975             if n >= 0.75:
976                 return 0.0
977
978             if n >= 0.5:
979                 continue
980
981             if n >= 0.75:
982                 return 0.0
983
984             if n >= 0.5:
985                 continue
986
987             if n >= 0.75:
988                 return 0.0
989
990             if n >= 0.5:
991                 continue
992
993             if n >= 0.75:
994                 return 0.0
995
996             if n >= 0.5:
997                 continue
998
999             if n >= 0.75:
1000                 return 0.0
1001
1002             if n >= 0.5:
1003                 continue
1004
1005             if n >= 0.75:
1006                 return 0.0
1007
1008             if n >= 0.5:
1009                 continue
1010
1011             if n >= 0.75:
1012                 return 0.0
1013
1014             if n >= 0.5:
1015                 continue
1016
1017             if n >= 0.75:
1018                 return 0.0
1019
1020             if n >= 0.5:
1021                 continue
1022
1023             if n >= 0.75:
1024                 return 0.0
1025
1026             if n >= 0.5:
1027                 continue
1028
1029             if n >= 0.75:
1030                 return 0.0
1031
1032             if n >= 0.5:
1033                 continue
1034
1035             if n >= 0.75:
1036                 return 0.0
1037
1038             if n >= 0.5:
1039                 continue
1040
1041             if n >= 0.75:
1042                 return 0.0
1043
1044             if n >= 0.5:
1045                 continue
1046
1047             if n >= 0.75:
1048                 return 0.0
1049
1050             if n >= 0.5:
1051                 continue
1052
1053             if n >= 0.75:
1054                 return 0.0
1055
1056             if n >= 0.5:
1057                 continue
1058
1059             if n >= 0.75:
1060                 return 0.0
1061
1062             if n >= 0.5:
1063                 continue
1064
1065             if n >= 0.75:
1066                 return 0.0
1067
1068             if n >= 0.5:
1069                 continue
1070
1071             if n >= 0.75:
1072                 return 0.0
1073
1074             if n >= 0.5:
1075                 continue
1076
1077             if n >= 0.75:
1078                 return 0.0
1079
1080             if n >= 0.5:
1081                 continue
1082
1083             if n >= 0.75:
1084                 return 0.0
1085
1086             if n >= 0.5:
1087                 continue
1088
1089             if n >= 0.75:
1090                 return 0.0
1091
1092             if n >= 0.5:
1093                 continue
1094
1095             if n >= 0.75:
1096                 return 0.0
1097
1098             if n >= 0.5:
1099                 continue
1100
1101             if n >= 0.75:
1102                 return 0.0
1103
1104             if n >= 0.5:
1105                 continue
1106
1107             if n >= 0.75:
1108                 return 0.0
1109
1110             if n >= 0.5:
1111                 continue
1112
1113             if n >= 0.75:
1114                 return 0.0
1115
1116             if n >= 0.5:
1117                 continue
1118
1119             if n >= 0.75:
1120                 return 0.0
1121
1122             if n >= 0.5:
1123                 continue
1124
1125             if n >= 0.75:
1126                 return 0.0
1127
1128             if n >= 0.5:
1129                 continue
1130
1131             if n >= 0.75:
1132                 return 0.0
1133
1134             if n >= 0.5:
1135                 continue
1136
1137             if n >= 0.75:
1138                 return 0.0
1139
1140             if n >= 0.5:
1141                 continue
1142
1143             if n >= 0.75:
1144                 return 0.0
1145
1146             if n >= 0.5:
1147                 continue
1148
1149             if n >= 0.75:
1150                 return 0.0
1151
1152             if n >= 0.5:
1153                 continue
1154
1155             if n >= 0.75:
1156                 return 0.0
1157
1158             if n >= 0.5:
1159                 continue
1160
1161             if n >= 0.75:
1162                 return 0.0
1163
1164             if n >= 0.5:
1165                 continue
1166
1167             if n >= 0.75:
1168                 return 0.0
1169
1170             if n >= 0.5:
1171                 continue
1172
1173             if n >= 0.75:
1174                 return 0.0
1175
1176             if n >= 0.5:
1177                 continue
1178
1179             if n >= 0.75:
1180                 return 0.0
1181
1182             if n >= 0.5:
1183                 continue
1184
1185             if n >= 0.75:
1186                 return 0.0
1187
1188             if n >= 0.5:
1189                 continue
1190
1191             if n >= 0.75:
1192                 return 0.0
1193
1194             if n >= 0.5:
1195                 continue
1196
1197             if n >= 0.75:
1198                 return 0.0
1199
1200             if n >= 0.5:
1201                 continue
1202
1203             if n >= 0.75:
1204                 return 0.0
1205
1206             if n >= 0.5:
1207                 continue
1208
1209             if n >= 0.75:
1210                 return 0.0
1211
1212             if n >= 0.5:
1213                 continue
1214
1215             if n >= 0.75:
1216                 return 0.0
1217
1218             if n >= 0.5:
1219                 continue
1220
1221             if n >= 0.75:
1222                 return 0.0
1223
1224             if n >= 0.5:
1225                 continue
1226
1227             if n >= 0.75:
1228                 return 0.0
1229
1230             if n >= 0.5:
1231                 continue
1232
1233             if n >= 0.75:
1234                 return 0.0
1235
1236             if n >= 0.5:
1237                 continue
1238
1239             if n >= 0.75:
1240                 return 0.0
1241
1242             if n >= 0.5:
1243                 continue
1244
1245             if n >= 0.75:
1246                 return 0.0
1247
1248             if n >= 0.5:
1249                 continue
1250
1251             if n >= 0.75:
1252                 return 0.0
1253
1254             if n >= 0.5:
1255                 continue
1256
1257             if n >= 0.75:
1258                 return 0.0
1259
1260             if n >= 0.5:
1261                 continue
1262
1263             if n >= 0.75:
1264                 return 0.0
1265
1266             if n >= 0.5:
1267                 continue
1268
1269             if n >= 0.75:
1270                 return 0.0
1271
1272             if n >= 0.5:
1273                 continue
1274
1275             if n >= 0.75:
1276                 return 0.0
1277
1278             if n >= 0.5:
1279                 continue
1280
1281             if n >= 0.75:
1282                 return 0.0
1283
1284             if n >= 0.5:
1285                 continue
1286
1287             if n >= 0.75:
1288                 return 0.0
1289
1290             if n >= 0.5:
1291                 continue
1292
129
```

3 Success: scikit-learn user base

350 000 returning users

5 000 citations



■ Windows ■ Mac

■ Linux

■ industry ■ academia ■ other

3 Open source is infrastructure Everybody uses it everyday



3 Open source is infrastructure

Everybody uses it everyday

In scientific research

- R: paper published in 2000, 67 248 citations
- LAPACK: book published in 2016, 7 165 citations
- scikit-learn: paper published in 2011, 5 525 citations

In the industry

“Roads and Bridge”: Ford foundation report

Talk by Heather Miller www.youtube.com/watch?v=17yy5BwliTw

3 Open source is infrastructure Everybody uses it everyday

An aerial night photograph of a complex highway interchange with multiple levels and ramps. The roads are illuminated by bright orange and yellow lights from the moving vehicles, creating a sense of motion. The surrounding city buildings are also lit up, with numerous billboards and signs visible. A large, semi-transparent white speech bubble is positioned in the center of the image, containing the text "It needs maintenance".

It needs maintenance

Recipe for good software:

Make it work, make it right, make it boring

3 The tragedy of the commons

Individuals, acting independently and rationally according to each one's self-interest, behave contrary to the whole group's long-term best interests by depleting some common resource.

Wikipedia



Make it work, make it right, make it boring

Core projects (boring) taken for granted
⇒ Hard to fund, less excitement

They need citation, in papers & on corporate web pages

3 The tragedy of the commons

Individuals, acting independently and rationally according to each one's self-interest, behave contrary to the whole group's long-term best interests by depleting some common resource.

Wikipedia

- + It's so hard to scale
- User support
- Growing codebase

Make it work, make it right, make it boring

Core projects (boring) taken for granted
⇒ Hard to fund, less excitement

They need citation, in papers & on corporate web pages

3 The tragedy of the commons

Individuals, acting independently and rationally according to each one's self-interest, behave contrary to the whole group's long-term best interests by depleting some common resource.

Wikipedia

- + It's so hard to scale
- User support
- Growing codebase

Make it work, make it right, make it boring

Core projects (boring) taken for granted
⇒ Hard to fund, less excitement

They need citation, in papers & on corporate web pages

3 The tragedy of the commons



Make it

Core project

⇒ Hard

They need c

G Varoquaux



Direction des affaires juridiques

Affaire suivie par :

[REDACTED] – juriste

Tel : [REDACTED]

[REDACTED]@anr.fr

INRIA Saclay
1 rue Honoré d'Estienne d'Orves
Campus de l'Ecole Polytechnique
Bâtiment Alan Turing
91120 PALAISEAU

Monsieur Braunschweig
Directeur de l'INRIA

Paris, le 9 déc. 2016

Nos Réf. : LabCom LearnClues (ANR-14-LAB6-0006 LEARNCLUES)
LRAR

Objet : Non validation du contrat de LabCom – arrêt du projet

Monsieur le Directeur,

Dans le cadre de l'Appel à Projets LabCom (*Laboratoires Communs / Organismes de Recherche – PME/ETI*) de l'édition 2014, le projet LearnClues, consistant en la mise en place d'un laboratoire commun entre l'INRIA Saclay (équipe Parietal) et la société TinyClues, a été sélectionné par l'ANR.

Ce partenariat a pour objectif principal l'évolution et l'amélioration du logiciel *sickit-learn* et de la bibliothèque y associée d'orchestration des calculs *Joblib*, créées par l'équipe Parietal, utilisés par ailleurs par la société TinyClues dans son activité commerciale.

L'INRIA a signé à ce titre une convention attributive n° ANR 14-LAB6-0006-01 en date du 9 décembre 2014 qui matérialise ses engagements et obligations et a désigné M. Gaël Varoquaux responsable scientifique.

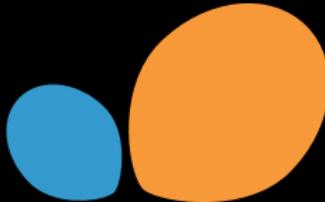
Aux termes de cette convention attributive (ainsi que de l'appel à projets Labcom), il est nécessaire de distinguer la phase de mise en place du laboratoire commun (« phase de montage ») et la phase



oring

web pages

Scikit-learn



The vision

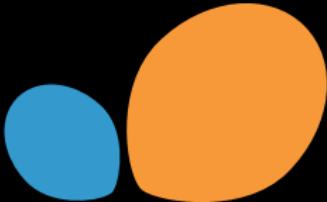
Make machine learning accessible

Versatile library: the “right” level of abstraction

Close to research, but seeking different tradeoffs



Scikit-learn



The vision

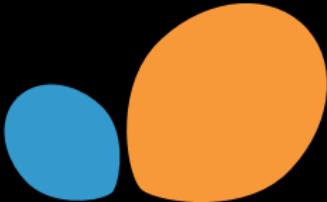
Make machine learning accessible

The tool

Scientific Python ecosystem
Simple API uniform across learners



Scikit-learn



The vision

Make machine learning accessible

The tool

Scientific Python ecosystem

The project

Many people working together
Tests and discussions for quality



Sustainability? A consortium?