



# Navigate machine learning

Victoriya Fedotova

With InTEL®

# Machine Learning: Your Path to Deeper Insight

Driving increasing innovation and commercial

Solutions

for reference across industries

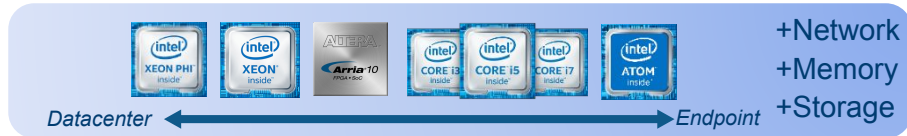
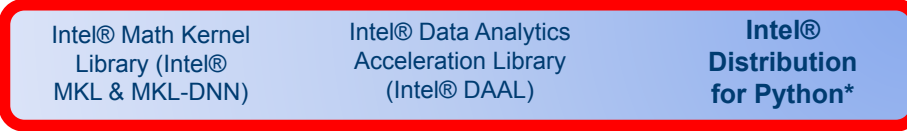
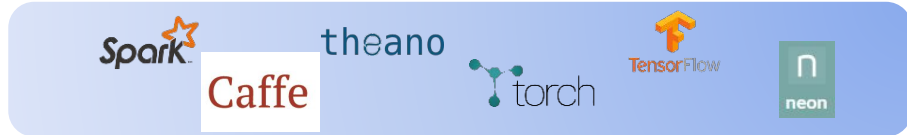
Tools/Platforms  
to accelerate deployment

Optimized Frameworks  
to simplify development

Libraries/Languages  
featuring optimized building blocks

Hardware Technology  
portfolio that is broad and cross-compatible

  
strategy provides the foundation for success using AI



+Network  
+Memory  
+Storage

# Motivation

Python is among the most popular programming languages

## Challenge #1:

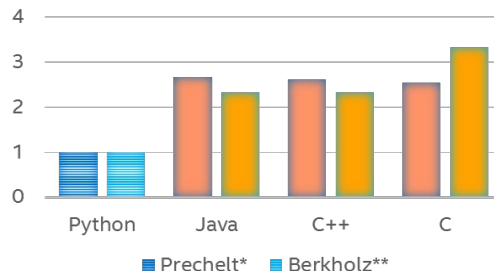
Domain specialists are not professional software programmers

## Challenge #2:

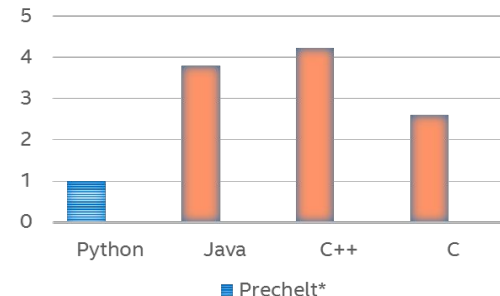
Python performance limits migration to production systems

- Hire a team of Java/C++ programmers ...
- OR
- Have team of Python programmers to deploy optimized Python in production

**LANGUAGE EXPRESSIVENESS**  
(LOC/FEATURE)



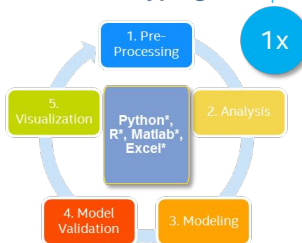
**PROGRAMMING COMPLEXITY (HOURS)**



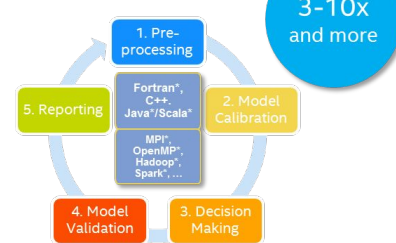
\* L.Prechelt, An empirical comparison of seven programming languages, IEEE Computer, 2000, Vol. 33, Issue 10, pp. 23-29  
\*\* RedMonk - D.Berkholz, Programming languages ranked by expressiveness



**Prototyping** Development cost



**Production** Development cost



3-10x and more

# Intel® Distribution for Python\*

Advancing Python performance closer to native speeds

**Easy, out-of-the-box  
access to high  
performance Python**

- Prebuilt, optimized for numerical computing, data analytics, HPC
- Drop in replacement for your existing Python. No code changes required

**High performance with  
multiple optimization  
techniques**

- Accelerated NumPy\*/SciPy\*/Scikit-Learn\* with Intel® MKL
- Data analytics with pyDAAL, enhanced thread scheduling with TBB, Jupyter\* Notebook interface, Numba\*, Cython\*
- Scale easily with optimized MPI4Py and Jupyter notebooks

**Faster access to latest  
optimizations for Intel  
architecture**

- Distribution and individual optimized packages available through conda and Anaconda Cloud: [anaconda.org/intel](https://anaconda.org/intel)
- Optimizations upstreamed back to main Python trunk

# Performance Gain from MKL (Compare

## Linear Algebra

- **BLAS**
- **LAPACK**
- ScaLAPACK
- Sparse BLAS
- Sparse Solvers

Up to 100x faster

## Fast Fourier Transforms

- **Multidimensional**
- FFTW interfaces
- Cluster FFT

Up to 10x faster!

## Vector Math

- **Trigonometric**
- **Hyperbolic**
- **Exponential**
- **Log**
- **Power, Root**

Up to 10x faster!

## Vector RNGs

- **Multiple BRNG**
- **Support methods for independent streams creation**
- **Support all key probability distributions**

Up to 60x faster!

## Summary Statistics

- Kurtosis
- Variation coefficient
- Order statistics
- Min/max
- Variance-covariance

## And More

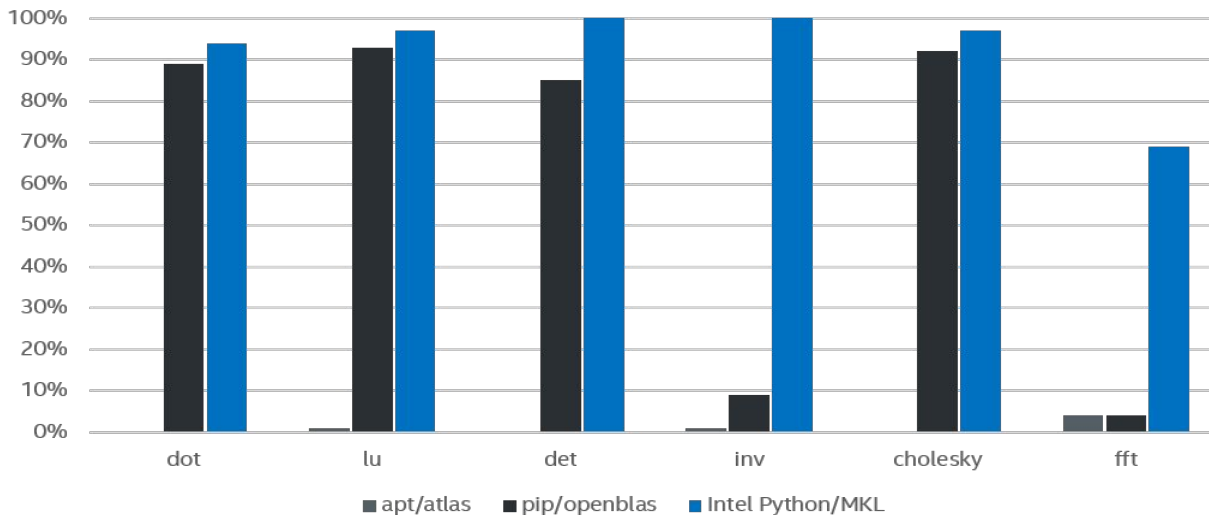
- Splines
- Interpolation
- Trust Region
- Fast Poisson Solver

Configuration info: - Versions: Intel® Distribution for Python 2017 Beta, icc 15.0; Hardware: Intel® Xeon® CPU E5-2698 v3 @ 2.30GHz (2 sockets, 16 cores each, HT=OFF), 64 GB of RAM, 8 DIMMS of 8GB@2133MHz; Operating System: Ubuntu 14.04 LTS.

# Out-of-the-box Performance with Intel® Distribution for Python\*

Mature AVX2 instructions based product

Python\* Performance as a Percentage of C/Intel® MKL for Intel® Xeon® Processors, 32 Core (Higher is Better)



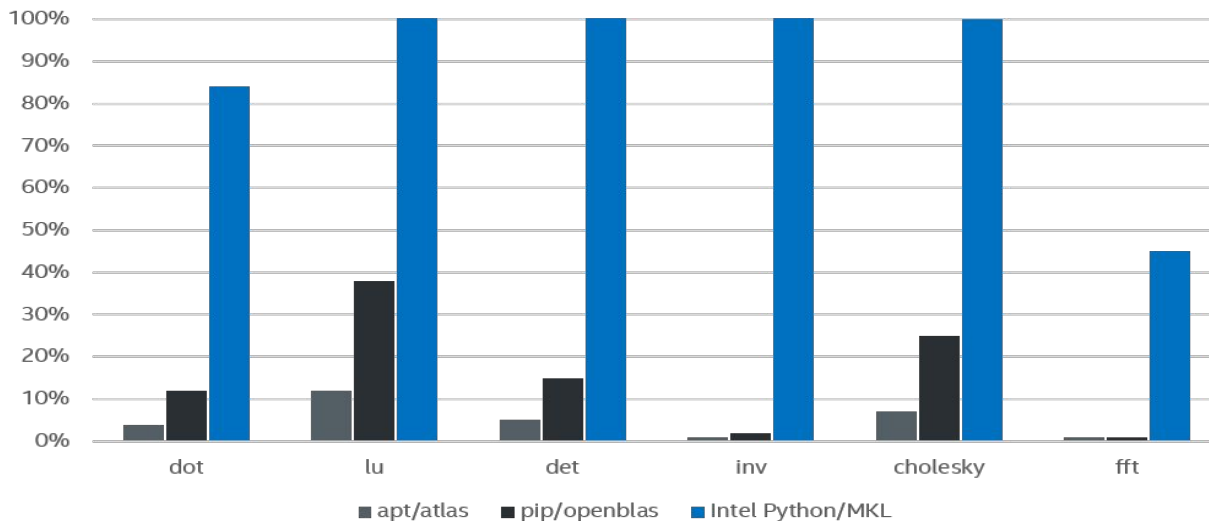
Configuration Info: apt/atlas: installed with apt-get, Ubuntu 16.10, python 3.5.2, numpy 1.11.0, scipy 0.17.0; pip/openblas: installed with pip, Ubuntu 16.10, python 3.5.2, numpy 1.11.1, scipy 0.18.0; Intel Python: Intel Distribution for Python 2017

Hardware: Xeon: Intel Xeon CPU E5-2698 v3 @ 2.30 GHz (2 sockets, 16 cores each, HT=off), 64 GB of RAM, 8 DIMMS of 8GB@2133MHz

# Out-of-the-box Performance with Intel® Distribution for Python\*

New AVX512 instructions based product

Python\* Performance as a Percentage of C/Intel® MKL for Intel® Xeon Phi™ Product Family, 64 Core (Higher is Better)



Configuration Info: apt/atlas: installed with apt-get, Ubuntu 16.10, python 3.5.2, numpy 1.11.0, scipy 0.17.0; pip/openblas: installed with pip, Ubuntu 16.10, python 3.5.2, numpy 1.11.1, scipy 0.18.0; Intel Python: Intel Distribution for Python 2017

Hardware: Intel Intel® Xeon Phi™ CPU 7210 1.30 GHz, 96 GB of RAM, 6 DIMMS of 16GB@1200MHz

# WORKSHOP

:

:

# BASIC functions



# Examples of Basic Functions

## NumPy, SciPy

- Matrix multiplication
- Random number generation
- Vector Math
- Linear algebra decompositions



matmul.py



matmul\_randintel.py



exp.py

## Not so basic functions

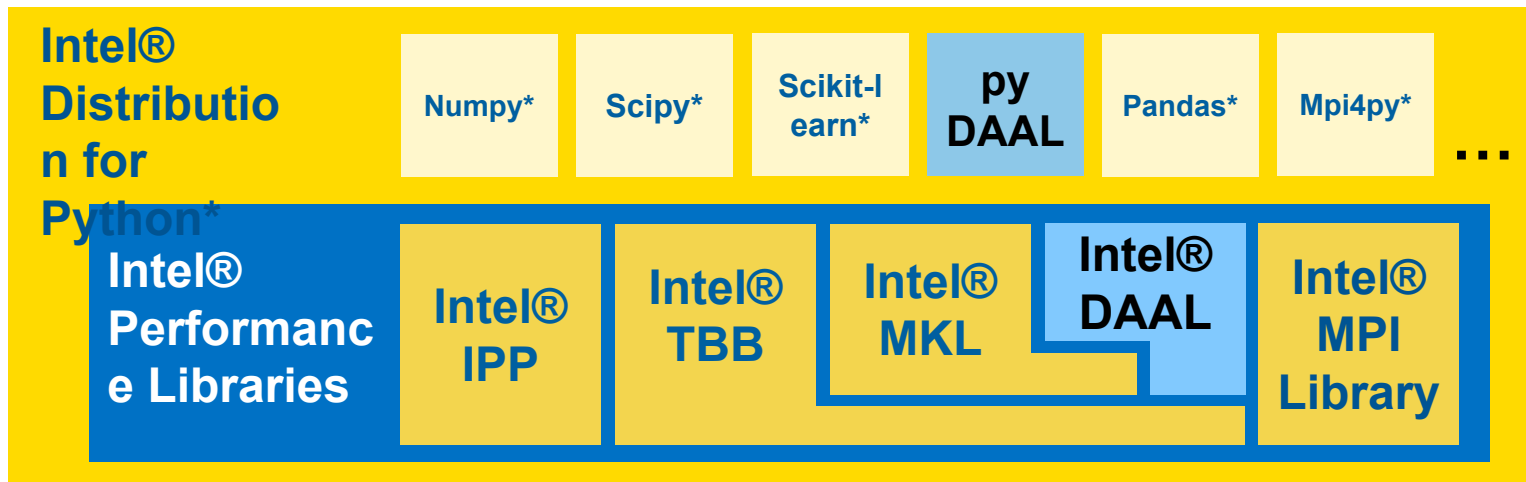
## SciKit-learn

- Linear regression
  - **NOTE:** Only Python 2.7 and 3.5 are supported for now



lr.py

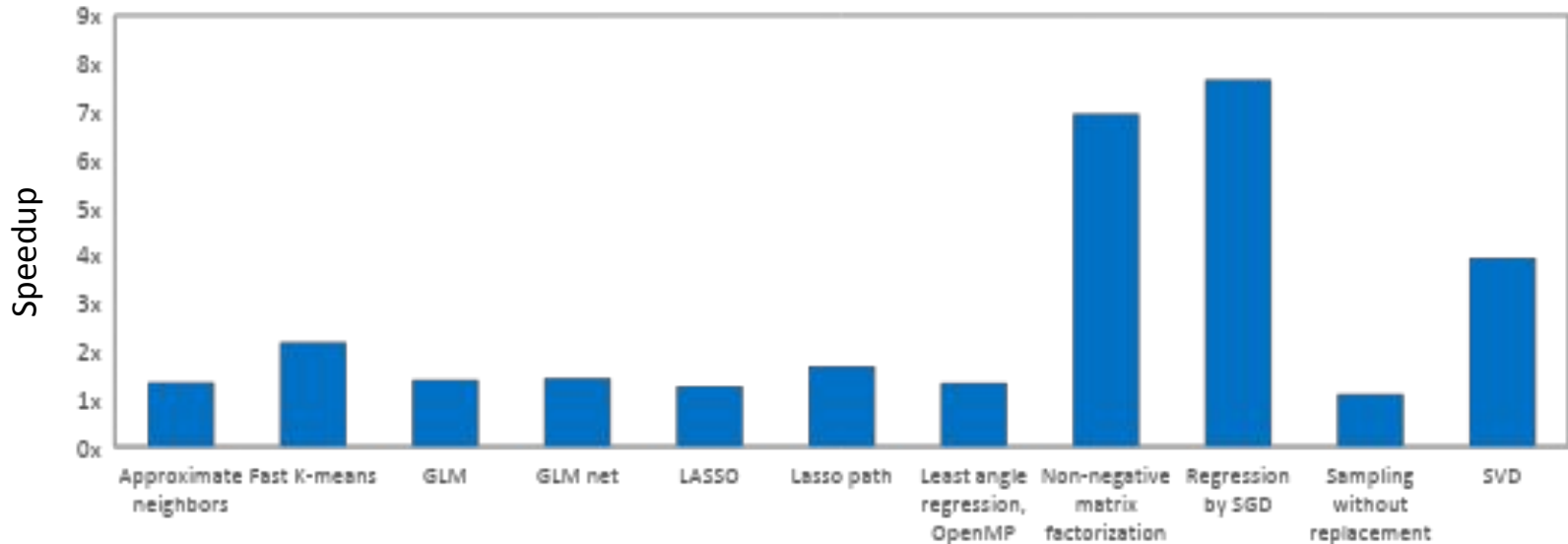
# Intel Python Landscape



# Scikit-Learn\* optimizations with Intel® MKL

## Speedups of Scikit-Learn\* Benchmarks (2017 Update 1)

Intel® Distribution for Python™ 2017 Update 1 vs. system Python & NumPy\*/Scikit-Learn\*

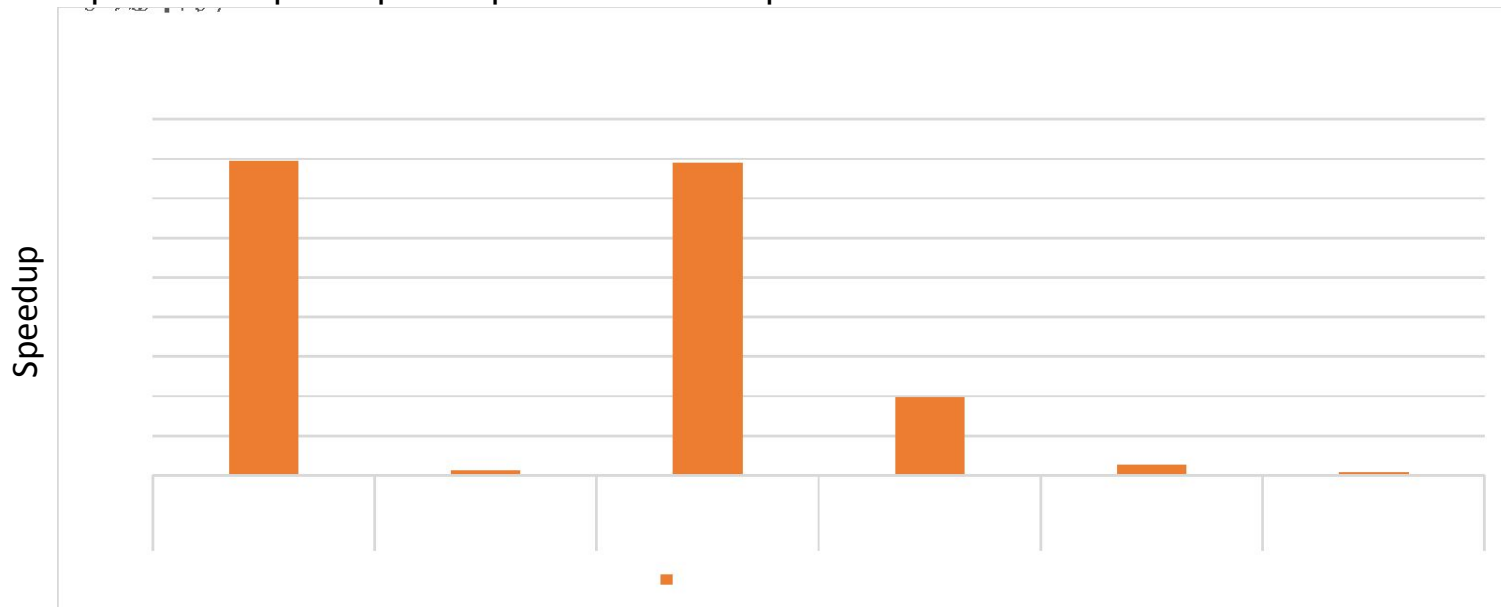


System info: 32x Intel® Xeon® CPU E5-2698 v3 @ 2.30GHz, disabled HT, 64GB RAM; Intel® Distribution for Python™ 2017 Gold; Intel® MKL 2017.0.0; Ubuntu 14.04.4 LTS; Numpy 1.11.1; scikit-learn 0.17.1. See Optimization Notice.

# More Scikit-Learn\* optimizations with Intel® DAAL

## Accelerated key Machine Learning algorithms with Intel® DAAL

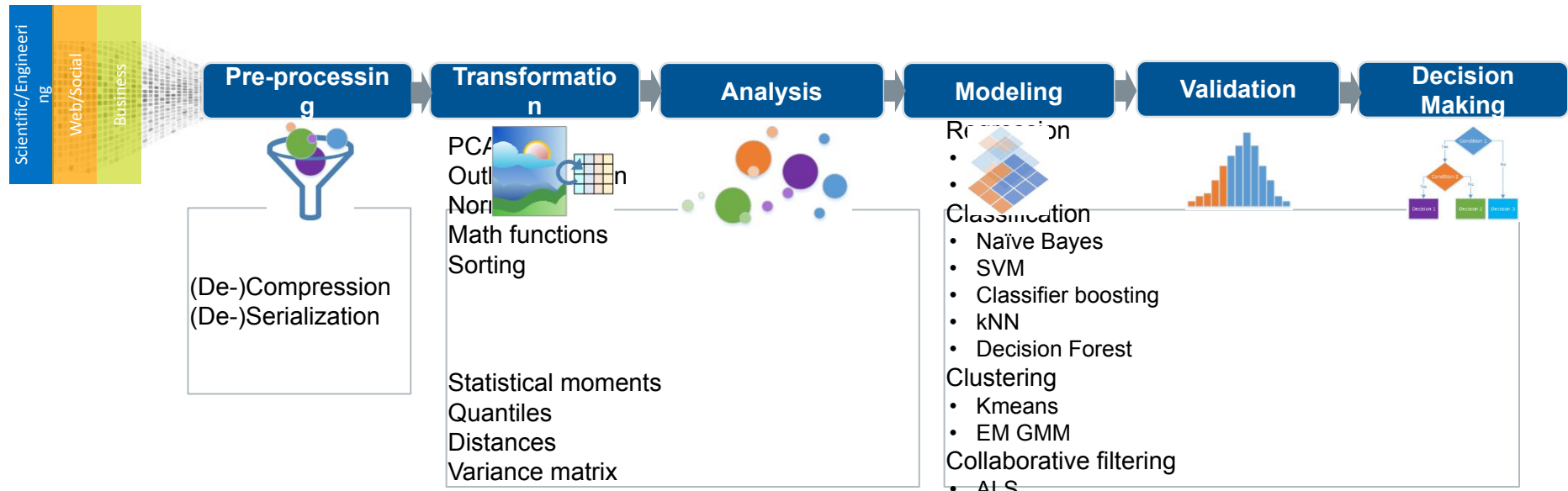
- Distances, K-means, Linear & Ridge Regression, PCA
- Up to 160x speedup on top of MKL initial optimizations



# Intel® DAAL: Heterogeneous Analytics

Available also in open source:  
<https://software.intel.com/en-us/articles/opendaal>

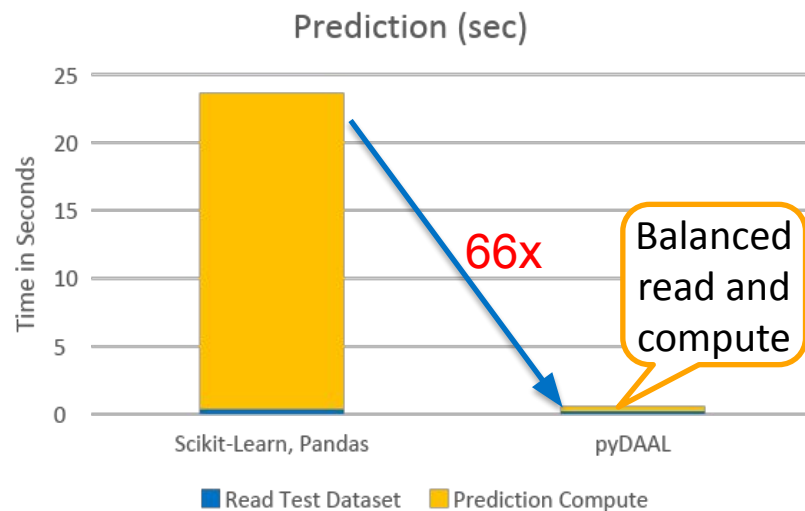
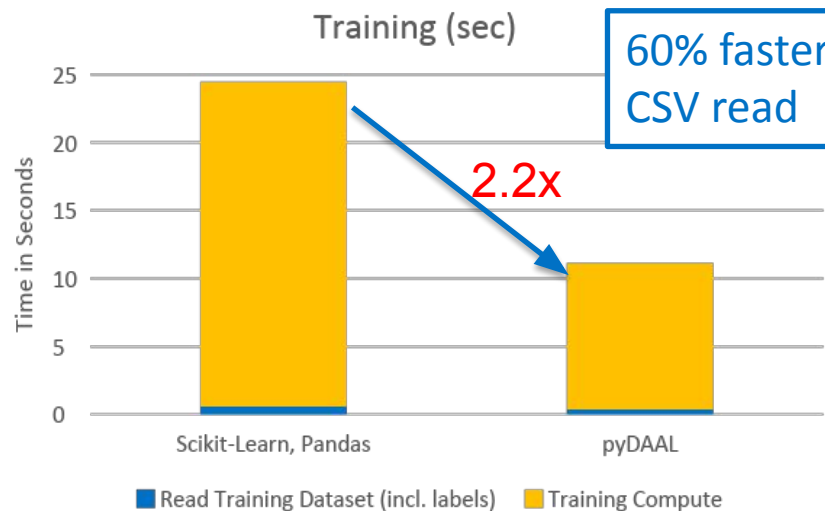
- Targets both data centers (Intel® Xeon® and Intel® Xeon Phi™) and edge-devices (Intel® Atom™)
- Perform analysis close to data source (sensor/client/server) to optimize response latency, decrease network bandwidth utilization, and maximize security
- Offload data to server/cluster for complex and large-scale analytics



# Performance Example : Read And Compute

## SVM Classification with RBF kernel

- Training dataset: CSV file (PCA-preprocessed MNIST, 40 principal components)  $n=42000$ ,  $p=40$
- Testing dataset: CSV file (PCA-preprocessed MNIST, 40 principal components)  $n=28000$ ,  $p=40$



System Info: Intel® Xeon® CPU E5-2680 v3 @ 2.50GHz, 504GB, 2x24 cores, HT=on, OS RH7.2 x86\_64, Intel® Distribution for Python\* 2017 Update 1 (Python\* 3.5)

# WORKSHOP : PyDAAL

# pyDAAL Getting Started

<https://github.com/daaltces/pydaal-getting-started>

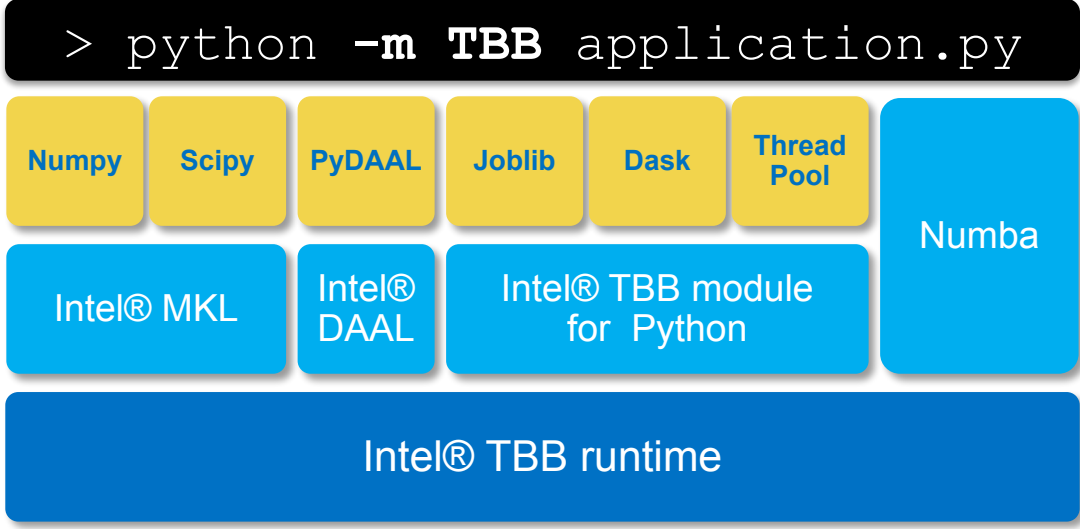
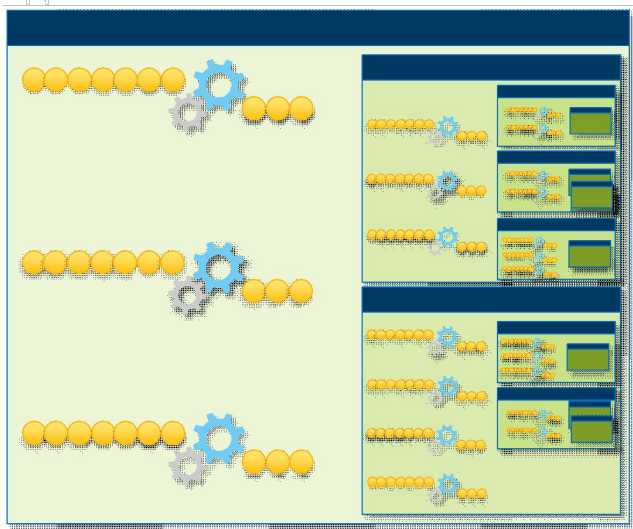
## DAAL4PY: Tech Preview

<https://software.intel.com/en-us/articles/daal4py-overview-a-high-level-python-api-to-the-intel-data-analytics-acceleration-library>



# Intel® TBB: parallelism orchestration in Python ecosystem

- Software components are built from smaller ones
- If each component is threaded there can be too much!
- Intel TBB dynamically balances thread loads and effectively manages oversubscription



# Profiling Python\* code with Intel® VTune™ Amplifier

Right tool for high performance application profiling at all levels

- Function-level and line-level hotspot analysis, down to disassembly
- Call stack analysis
- Low overhead
- Mixed-language, multi-threaded application analysis

Feature	cProfile	Line_profiler	Intel® VTune™ Amplifier
Profiling technology	Event	Instrumentation	Sampling, hardware events
Analysis granularity	Function-level	Line-level	Line-level, call stack, time windows, hardware events
Intrusiveness	Medium (1.3-5x)	High (4-10x)	Low (1.05-1.3x)
Mixed language programs	Python	Python	Python, Cython, C++, Fortran

# Installing Intel® Distribution for Python\* 2017

Stand-alone installer and [anaconda.org/intel](https://anaconda.org/intel)

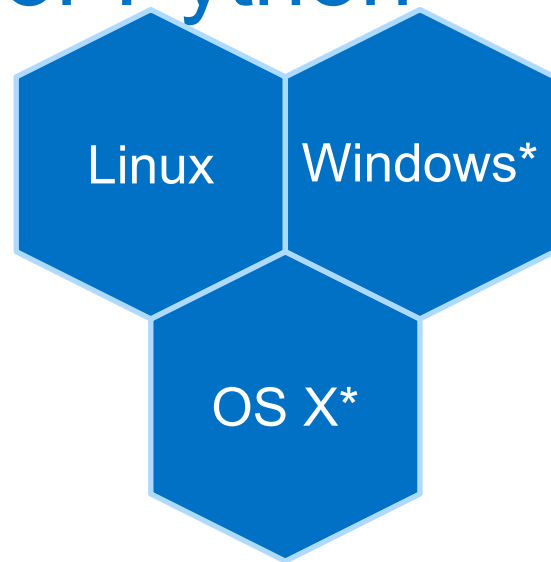
Download full installer from

<https://software.intel.com/en-us/intel-distribution-for-python>

OR

```
> conda config --add channels intel  
> conda install intelpython3_full  
> conda install intelpython3_core
```

```
docker pull intelpython/intelpython3_full
```



# Intel® Distribution for Python




























<https://software.intel.com/en-us/distribution-for-python>

# backup

# Collaborative Filtering

- Processes users' past behavior, their activities and ratings
- Predicts, what user might want to buy depending on his/her preferences


























Collaborative Filtering

From Wikipedia

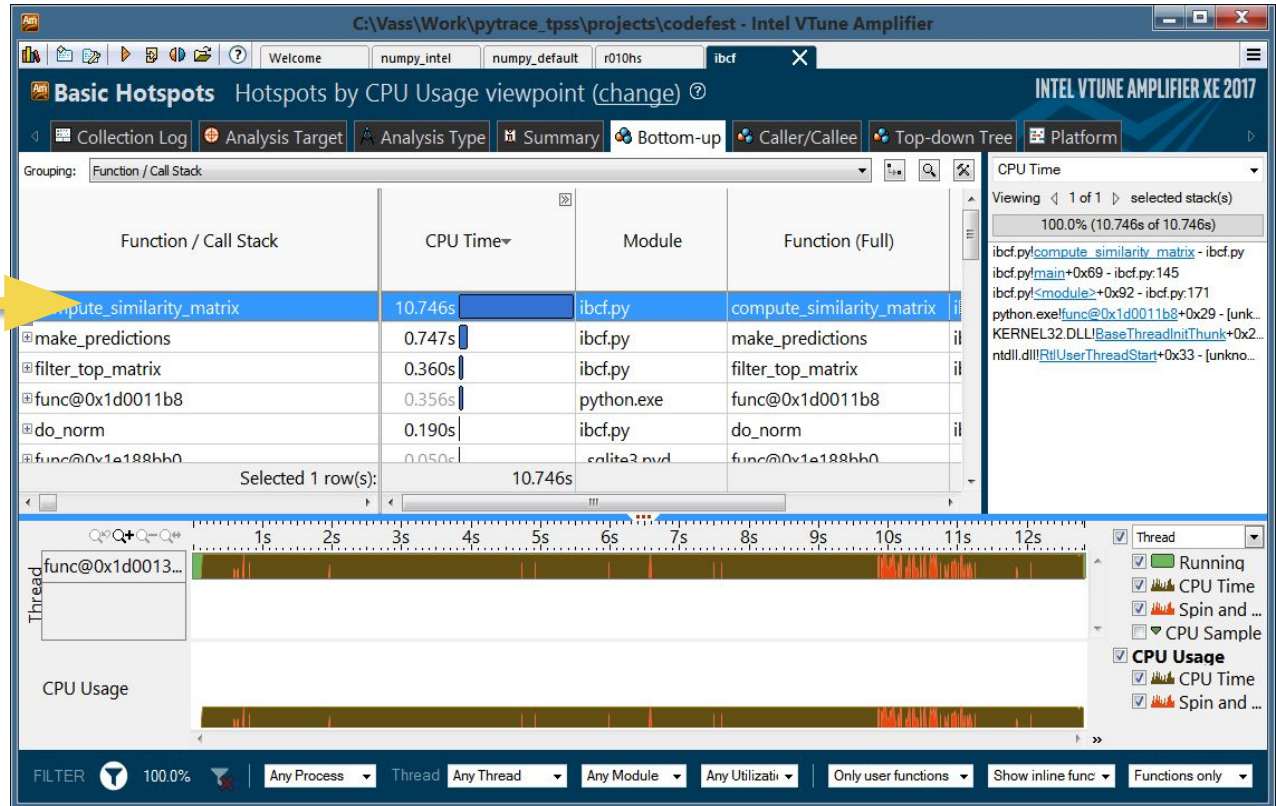


Similarities in users preferences (in Green) are used to predict ratings

# Training: Profiling pure python\*

Items similarity assessment (similarity matrix computation) is the main hotspot

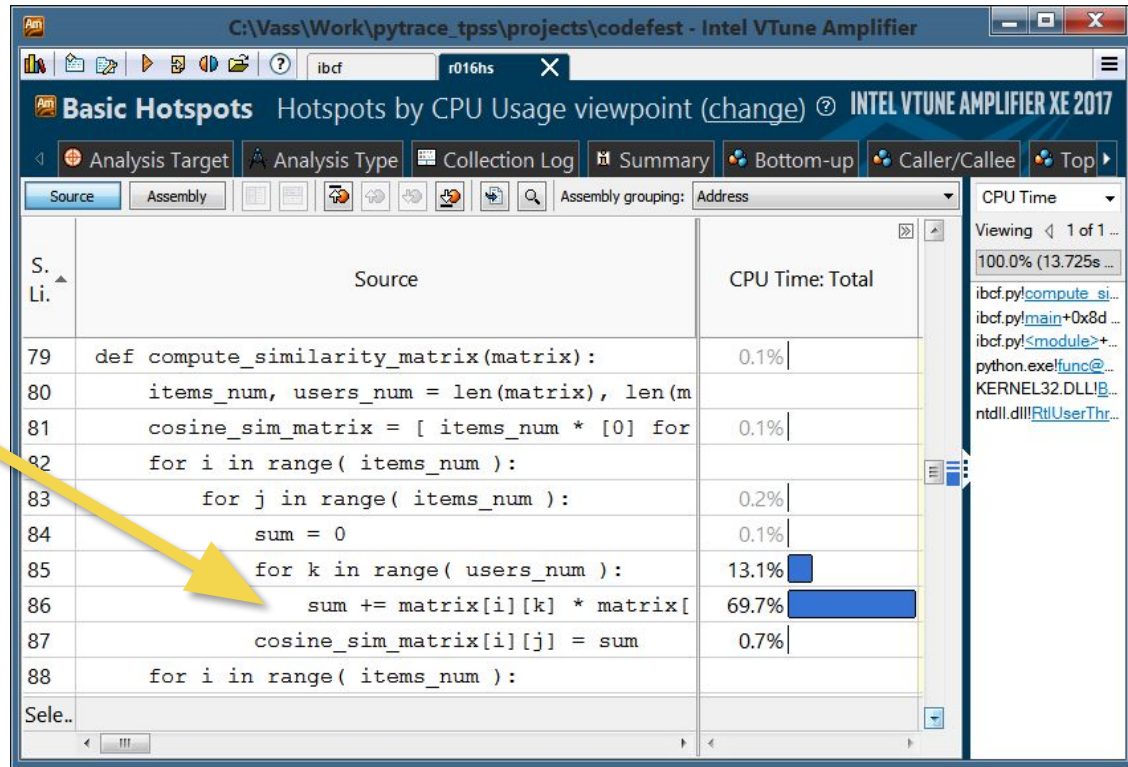


Configuration Info: - Versions: Red Hat Enterprise Linux\* built Python\*: Python 2.7.5 (default, Feb 11 2014), NumPy 1.7.1, SciPy 0.12.1, multiprocessing 0.70a1 built with gcc 4.8.2; Hardware: 24 CPUs (HT ON), 2 Sockets (6 cores/socket), 2 NUMA nodes, Intel(R) Xeon(R) X5680@3.33GHz, RAM 24GB, Operating System: Red Hat Enterprise Linux Server release 7.0 (Maipo)

# Training: Profiling pure Python\*

This loop is major bottleneck. Use appropriate technologies (NumPy/SciPy/Scikit-Learn or Cython/Numba) to accelerate

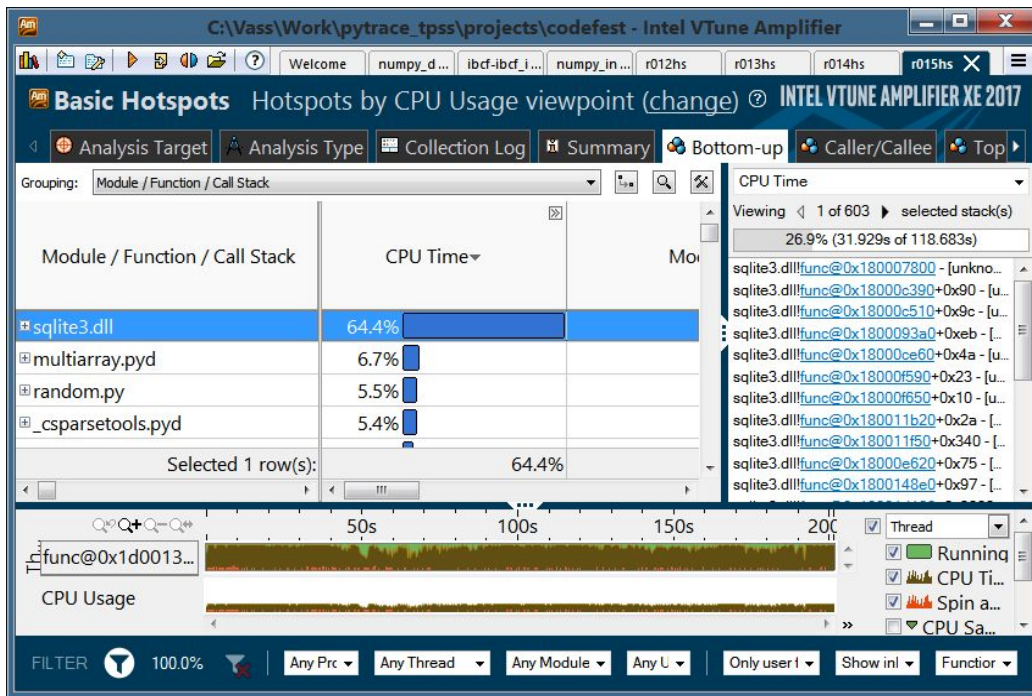
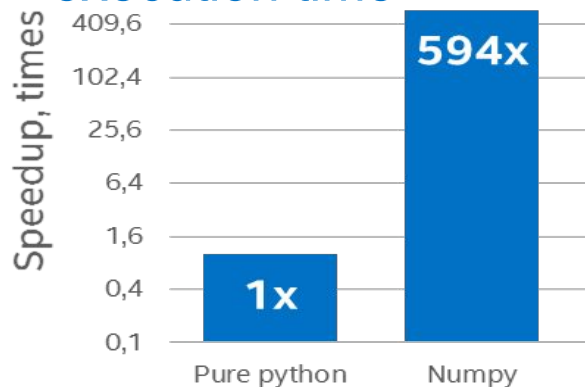
Configuration Info: - Versions: Red Hat Enterprise Linux\* built Python\*: Python 2.7.5 (default, Feb 11 2014), NumPy 1.7.1, SciPy 0.12.1, multiprocessing 0.70a1 built with gcc 4.8.2; Hardware: 24 CPUs (HT ON), 2 Sockets (6 cores/socket), 2 NUMA nodes, Intel(R) Xeon(R) X5680@3.33GHz, RAM 24GB, Operating System: Red Hat Enterprise Linux Server release 7.0 (Maipo)





# Training: Python + Numpy (MKL)

- Much faster!
- The most compute-intensive part takes ~5% of all the execution time



Configuration info: 96 CPUs (HT ON), 4 Sockets (12 cores/socket), 1 NUMA nodes, Intel(R) Xeon(R) E5-4657L v2@2.40GHz, RAM 64GB, Operating System: Fedora

© 2017 Intel Corporation. All rights reserved. Intel and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

For more complete information about compiler optimizations, see our [Optimization Notice](#).

# Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

For more complete information about compiler optimizations, see our Optimization Notice at <https://software.intel.com/en-us/articles/optimization-notice#opt-en>.

Copyright © 2017, Intel Corporation. All rights reserved. Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries. \*Other names and brands may be claimed as the property of others.



Software