USING INTEL DAAL

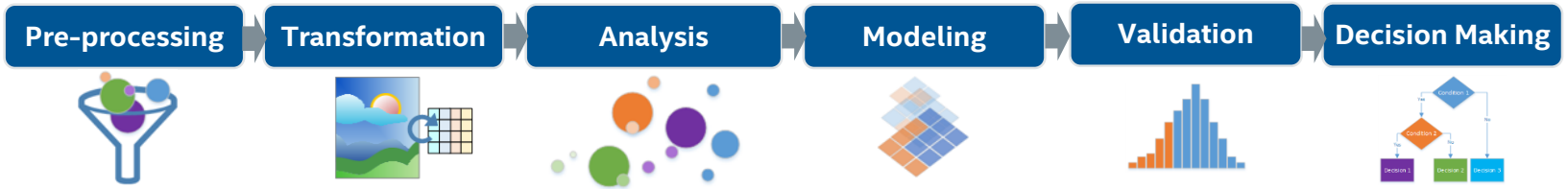
Walter Riviera

EMEA AI- TSS

INTEL® DATA ANALYTICS ACCELERATION LIBRARY (INTEL® DAAL)

High Performance Machine Learning and Data Analytics Library

Building blocks for all data analytics stages, including data preparation, data mining & machine learning



Open Source • Apache 2.0 License

Common Python, Java and C++ APIs across all Intel hardware

Optimized for large data sets including streaming and distributed processing

Flexible interfaces to leading big data platforms including Spark and range of data formats (CSV, SQL, etc.)

INTEL DISTRIBUTION FOR PYTHON

Advancing Python* Performance Closer to Native Speeds

software.intel.com/intel-distribution-for-python

For developers using the most popular and fastest growing programming language for AI

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)

Drive 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
- Optimizations upstreamed back to main Python trunk

INSTALLING INTEL® DAAL

Through Anaconda Official repository (full collection IDP + DAAL + MKL):
https://www.anaconda.com/download/

Through “pip” command (single packages):
https://software.intel.com/en-us/articles/installing-the-intel-distribution-for-python-and-intel-performance-libraries-with-pip-and

Package Name	pip command	Platform Availability
numpy	*pip install intel-numpy*	Linux, Win, macOS(10.12)
scipy	*pip install intel-scipy*	
scikit-learn	*pip install intel-scikit-learn*	
pydaal	*pip install pydaal*	
tbb4py	*pip install tbb4py*	

REPLICATING THE DEMO EXAMPLE

```
# Download and install Anaconda as recommended on their official website:
# https://www.anaconda.com/download/

# Once Anaconda is up&running, type:
$ conda create -c intel --name my_environment pip     # This create a new python env. with its own pip in it

$ conda activate my_environment                       # which will allow us to enter the env.
(my_environment) $

# Now we have to install all the dependencies “D” required by the code example, by simply typing “pip install D”.
# Having created a Virtual environment with its own “pip” cmd, will allow us to keep the 2 environment (native
 and  Virtualized) separate and independents.

# Once all the dependencies have been installed within the virtual env, we can proceed with:
 (my_environment) $ pip install intel-scikit-learn

# to see the different execution time, we just have to run the same code outside and inside the virtual env.
# To deactivate the virtual environment and return to the native machine, we just need to run:
$ conda deactivate
```