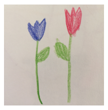
LALE Documentation

IBM AI Research

Dec 10, 2019

CONTENTS:



Lale is a Python library for semi-automated data science. Lale makes it easy to automatically select algorithms and tune hyperparameters of pipelines that are compatible with scikit-learn, in a type-safe fashion. If you are a data scientist who wants to experiment with automated machine learning, this library is for you! Lale adds value beyond scikit-learn along three dimensions: automation, correctness checks, and interoperability. For *automation*, Lale provides a consistent high-level interface to existing pipeline search tools including GridSearchCV, SMAC, and Hyperopt. For *correctness checks*, Lale uses JSON Schema to catch mistakes when there is a mismatch between hyperparameters and their type, or between data and operators. And for *interoperability*, Lale has a growing library of transformers and estimators from popular libraries such as scikit-learn, XGBoost, PyTorch etc. Lale can be installed just like any other Python package and can be edited with off-the-shelf Python tools such as Jupyter notebooks.

CHAPTER

ONE

INSTALLATION

1.1 Installing from PyPI

Lale is easy to install. Assuming you already have a Python 3.6+ environment, all you need is the following:

```
pip install lale
```

This will install the **Lale Core** setup target, which includes many operators, pipelines, and search space generation targeting hyperopt and scikit-learn's GridSearchCV. It has a smaller set of dependencies than the **Lale Full** setup target, which also includes search space generation for SMAC and some deep-learning operators. You can install it as follows:

```
pip install lale[full]
```

Now you should be ready to start using Lale, for instance, in a Jupyter notebook.

1.2 Installing from Source

As an alternative to installing Lale directly from the online github repository, you can also first clone the repository and then install Lale from your local clone. For the **Lale Core** setup target:

```
git clone https://github.com/IBM/lale.git
cd lale
pip install .
```

For the **Lale Full** and **Lale Test** setup targets:

```
pip install .[full,test]
```

Now, you are ready to run some tests. For a quick check, do the following in the `lale` directory:

```
export PYTHONPATH=`pwd`
python -m unittest test.test.TestLogisticRegression
```

The output should look like:

```
Ran 20 tests in 105.201s
OK
```

To run the full test suite, do the following in the `lale` directory:

```
make run_tests
```

1.3 Setting up the Environment

For the full functionality of Lale, you will need a Python 3.6+ environment, as well as g++, graphviz, make, and swig. You can use Lale on Linux, Windows 10, or Mac OS X. Depending on your operating system, you can skip ahead to the appropriate section below.

1.3.1 On Windows 10

First, you should enable the Windows Subsystem for Linux (WSL):

- Start/search -> Settings -> Update&Security -> For developers -> activate "Developer Mode"
- Start/search -> Turn Windows features on or off -> WSL -> restart
- Internet Explorer -> https://aka.ms/wslstore -> Ubuntu -> get
- Start/search -> Ubuntu

At this point, you can continue with the instructions in section *On Ubuntu Linux*.

1.3.2 On Ubuntu Linux

Start by making sure your Ubuntu installation is up-to-date and check the version. In a command shell, type:

```
sudo apt update
sudo apt upgrade
lsb_release -a
```

This should output something like "Description: Ubuntu 16.04.4 LTS".

Also, make sure you have g++, make, graphviz, and swig installed. Otherwise, you can install them:

```
sudo apt install g++
sudo apt install graphviz
sudo apt install make
sudo apt install swig
```

Next, set up a Python virtual environment with Python 3.6.

```
sudo add-apt-repository ppa:jonathonf/python-3.6
sudo apt update
sudo apt install python3.6
sudo apt install python3.6-dev
sudo apt install virtualenv
virtualenv -p /usr/bin/python3.6 ~/python3.6venv
source ~/python3.6venv/bin/activate
```

At this point, you can continue with the Lale *Installation* instructions at the top of this file.

1.3.3 On Mac OS X

Assuming you already have a Python 3.6+ virtual environment, you will need to install swig using brew before you can install Lale.

If you encounter any issues in installing SMAC:

MacOS 10.14

```
open /Library/Developer/CommandLineTools/Packages/macOS_SDK_headers_for_macOS_10.14.
↪pkg
```

Then

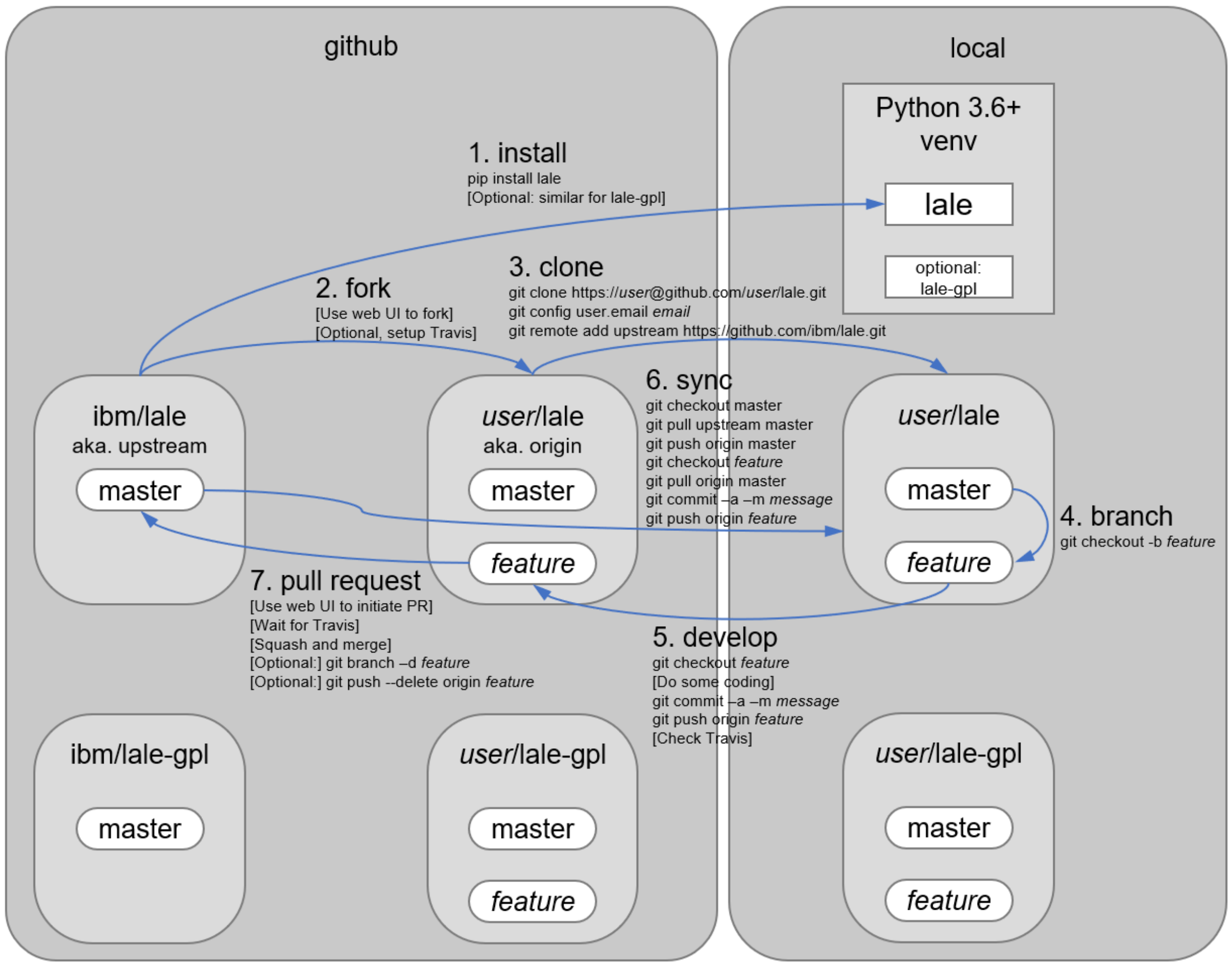
```
export CPATH=/Library/Developer/CommandLineTools/usr/include/c++/v1
```

MacOS 10.15 Catalina:

```
CFLAGS=-stdlib=libc++  pip install smac
```

1.4 For Developers

If you want to develop Lale itself, we recommend you create a fork with a feature branch. This is the typical workflow for open-source projects on github. In addition, we expect contributors to submit a "Developer's Certificate of Origin" by signing the following form: DCO1.1.txt. Below is a visualization of the workflow.



Italics in the visualization indicate parts you have to substitute: *user* (your user name), *email* (your email associated with github), *feature* (the name of the feature branch you are working on), and *message* (a description of your commit).

The name Lale, pronounced *laleh*, comes from the Persian word for tulip. Similarly to popular machine-learning libraries such as scikit-learn, Lale is also just a Python library, not a new stand-alone programming language. It does not require users to install new tools nor learn new syntax.

The following paper has a technical deep-dive:

```
@Article{arxiv19-lale,
author = "Hirzel, Martin and Kate, Kiran and Shinnar, Avraham and Roy, Subhrajit and␣
↪Ram, Parikshit",
title = "Type-Driven Automated Learning with {Lale}",
journal = "CoRR",
volume = "abs/1906.03957",
year = 2019,
month = may,
url = "https://arxiv.org/abs/1906.03957" }
```

Contributors are expected to submit a “Developer’s Certificate of Origin”, which can be found in DCO1.1.txt.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search