

Cloudera Data Science Workbench 1.10.0.0

Cloudera Data Science Workbench Runtimes

Date published: 2020-02-28

Date modified: 2021-08-05

CLUDERA

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Managing ML Runtimes.....	5
ML Runtimes versus Legacy Engine.....	5
Using Runtime Catalog.....	7
ML Runtimes Nvidia GPU Edition.....	7
Testing ML Runtime GPU Setup.....	8
ML Runtimes NVIDIA RAPIDS Edition.....	9
Using Editors for ML Runtimes.....	10
Using JupyterLab with ML Runtimes.....	10
Installing a Jupyter extension.....	11
Installing a Jupyter kernel.....	11
Using Conda Runtime.....	12
Installing Additional ML Runtimes Packages.....	13
Restrictions for upgrading R and Python packages.....	14
ML Runtimes Environment Variables.....	14
ML Runtimes Environment Variables List.....	15
Accessing Environmental Variables from Projects.....	16
Customized Runtimes.....	17
Creating Customized ML Runtimes.....	17
Create a Dockerfile for the Custom Runtime Image.....	18
Metadata for Custom ML Runtimes.....	18
Editor Customization.....	19
Build the New Docker Image.....	19
Distribute the Image.....	19
Add Docker registry credentials.....	20
Add the new ML Runtime.....	20
Limitations.....	21
Pre-Installed Packages in ML Runtimes.....	21
ML Runtimes 2023.05.....	21
Python 3.10 Libraries for Conda.....	21
Python 3.10 Libraries for Workbench.....	25
Python 3.9 Libraries for Workbench.....	30
Python 3.8 Libraries for Workbench.....	37
Python 3.7 Libraries for Workbench.....	45

Scala 2.11 Libraries for Workbench.....	52
Python 3.10 Libraries for JupyterLab.....	52
Python 3.9 Libraries for JupyterLab.....	59
Python 3.8 Libraries for JupyterLab.....	66
Python 3.7 Libraries for JupyterLab.....	73
R 4.0 Libraries.....	79
R 4.1 Libraries.....	82
R 3.6 Libraries.....	86
ML Runtimes 2021.09.....	90
Python 3.9 Libraries for Workbench.....	90
Python 3.8 Libraries for Workbench.....	92
Python 3.7 Libraries for Workbench.....	94
Python 3.6 Libraries for Workbench.....	97
Scala 2.11 Libraries for Workbench.....	99
Python 3.9.6 Libraries for JupyterLab.....	99
Python 3.8 Libraries for JupyterLab.....	104
Python 3.7 Libraries for JupyterLab.....	108
Python 3.6 Libraries for JupyterLab.....	112
R 4.0 Libraries.....	117
R 3.6 Libraries.....	119
ML Runtimes 2021.06.....	121
Python 3.8.6 Libraries for Workbench.....	121
Python 3.7.9 Libraries for Workbench.....	123
Python 3.6.12 Libraries for Workbench.....	125
Python 3.8.6 Libraries for JupyterLab.....	128
Python 3.7.9 Libraries for JupyterLab.....	131
Python 3.6.12 Libraries for JupyterLab.....	135
R 4.0 Libraries.....	139
R 3.6 Libraries.....	142
ML Runtimes 2021.04.....	144
RAPIDS Runtime PIP Python 3.7.8 Libraries for Workbench.....	144
RAPIDS Runtime PIP Python 3.8.6 Libraries for Workbench.....	148
RAPIDS Runtime PIP Python 3.7.8 Libraries for JupyterLab.....	153
RAPIDS Runtime PIP Python 3.8.6 Libraries for JupyterLab.....	157
ML Runtimes 2021.02.....	162
Python 3.8 Libraries for Workbench.....	162
Python 3.7 Libraries for Workbench.....	164
Python 3.6 Libraries for Workbench.....	166
Python 3.8 Libraries for JupyterLab.....	168
Python 3.7 Libraries for JupyterLab.....	171
Python 3.6 Libraries for JupyterLab.....	175
R 4.0 Libraries.....	179
R 3.6 Libraries.....	181
ML Runtimes 2020.11.....	184

Managing ML Runtimes

Provides overview, installation, set up, configuration, and customization information for Machine Learning Runtimes.

ML Runtimes are responsible for running the code written by users and intermediating access to the CDH cluster.

You can think of an ML Runtime as a virtual machine, customized to have all the necessary dependencies to access the computing cluster while keeping each project's environment entirely isolated. To ensure that every ML Runtime has access to the parcels and client configuration managed by the Cloudera Manager Agent, a number of folders are mounted from the host into the container environment.

ML Runtimes versus Legacy Engine

While Runtimes and the Legacy Engine are both container images that contain the Linux OS, interpreter(s), and libraries, ML Runtimes keeps the images small and improves performance, maintenance, and security.



Note: Legacy Engines are deprecated since June 2021 and will be removed in a future release. Starting with version 2.0.38version 1.5.1, on new workspaces Legacy Engines are disabled by default and no Legacy Engine image is registered in the workspace. Cloudera recommends using ML Runtimes for all new projects, and urges customers to migrate existing Engine-based projects to ML Runtimes.

Runtimes and the Legacy Engine serve the same basic goal: they are container images that contain a complete Linux OS, interpreter(s), and libraries. They are the environment in which your code runs. However, ML Runtimes design keeps the images small, which improves performance, maintenance, and security.

There is one Legacy Engine. The Engine is monolithic. It contains the machinery necessary to run sessions using all four Engine interpreter options that Cloudera currently supports (Python 2, Python 3, R, and Scala) and a much larger set of UNIX tools including LaTeX. The Conda package manager was available in the Legacy Engine. Conda is not available in ML Runtimes.

Runtimes are the future of CML. There are many Runtimes. Currently each Runtime contains a single interpreter (for example, Python 3.8, R 4.0) and a set of UNIX tools including `gcc`. Each Runtime supports a single UI for running code (for example, the Workbench or JupyterLab).

To migrate from Legacy Engine to Runtimes, you'll need to modify your project settings. See *Modifying Project Settings* for more information.

Disable Engines

Starting with version 2.0.38version 1.5.1, Legacy Engines are disabled by default on the workspace level. This setting is accessible in `Site Administration Runtime Disable Engines`. Select this checkbox on upgraded workspaces to disable Legacy Engines and change all existing project types to ML Runtime. Disabling Legacy Engines prohibits changing the project type to Legacy Engine.

This has the following effect for workloads using Legacy Engine kernels:

- Model builds cannot be redeployed
- Applications cannot be restarted
- Jobs cannot be started, including any kind of scheduled Jobs
- Jobs cannot be started, scheduled jobs will fail
- Sessions cannot be started

To re-enable Legacy Engines go to `Site Administration Runtime` and deselect the `Disable Engines` checkbox under `Engine Images`.

Jupyter

Our Python Runtimes support JupyterLab, a general purpose IDE from the Jupyter project. The engine supports Jupyter Notebook, a simpler UI focused on Notebooks. If you prefer the simpler Notebook UI, choose `Classic`

Notebook from the JupyterLab Help menu. To further customize the JupyterLab experience on CML see *Using Editors for ML Runtimes*.

Build dependencies

Runtimes generally include fewer UNIX tools than the Legacy Engine. This means you are more likely to find that you cannot install a Python or R package because the Runtime is missing a build dependency such as a library. This should not happen often with Python. Most Python packages are distributed as precompiled “wheels”, so there are no build dependencies. It is more likely to happen with R packages because precompiled packages are not available for our architecture. We have tried to cover most common use cases, but if you find you cannot build something, then please contact customer support.

Using pip to install libraries in Python

To install a Python library from within Workbench or JupyterLab we recommend you use `%pip` (for example, `%pip install sklearn`). `%pip` is a “magic” command that is guaranteed to point to the right version of pip. This is a good habit to get into, as it will work outside CML. Note you do not need to add “3” to install a Python 3 library.

If you prefer to use the pip executable directly, both `pip` and `pip3` work. This is because Runtimes do not include Python 2. Like any shell command, precede it with “!” to run it from within Workbench or JupyterLab (for example, `!pip install sklearn`). In the Legacy Engine you must use `pip3` to install Python 3 packages and the `%pip` magic command is not supported.

Python paths

Python Runtimes include preinstalled Python packages at `/usr/local/lib/python/<version>/site-packages`. The pre-installed packages and versions are documented in *Pre-Installed Packages in ML Runtimes*.

When you use pip, you install packages into the current project (not a runtime image) at `/home/cdsw/.local/lib/python/<version>/site-packages`. This means you need to reinstall packages if you change Python versions.

In most cases, you can install a newer version of a package preinstalled in `/usr/local` into your project. For example, we preinstall `numpy` and you can install a newer version. But there are some exceptions to this: if you install `matplotlib`, `ipykernel`, or its dependencies (`ipython`, `traitlets`, `jupyter_client`, and `tornado`) then you may break your ability to launch sessions.

If you accidentally install these packages (or you see unexpected behavior when you switch a project from Legacy Engine to Runtimes), the simplest solution is to delete `/home/cdsw/.local/lib/python` and reinstall your project’s dependencies from the project overview page.

R paths

R Runtimes include preinstalled R packages at `/usr/local/lib/R/library/`. The pre-installed packages and versions are documented in *Pre-Installed Packages in ML Runtimes*.

When you use `install.packages()`, you install packages into the current project (not a runtime image) at `/home/cdsw/.local/lib/R/<version>/library` (for example, `$R_LIBS_USER`). This means you need to reinstall packages if you change R versions.

Note the R project package path in Legacy Engines. If you use engines, you install packages to `/home/cdsw/R`. The change to `/home/cdsw/.local/lib/R/<version>/library` was made to support multiple versions of R.

In most cases, you can install a newer version of a package preinstalled `/usr/local` into your project. For example, we preinstall `ggplot2` and you can install a newer version. But there are two exceptions to this. If you install `Cairo` or `Rserve` they may break your ability to launch sessions.

If you accidentally install these packages (or you see unexpected behavior when you switch a project from Legacy Engine to Runtimes), the simplest solution is to delete `/home/cdsw/.local/lib/python` and reinstall your project’s dependencies from the project overview page.

Related Information

[Creating a Project with Legacy Engine Variants](#)

[Modifying Project Settings](#)

Using Runtime Catalog

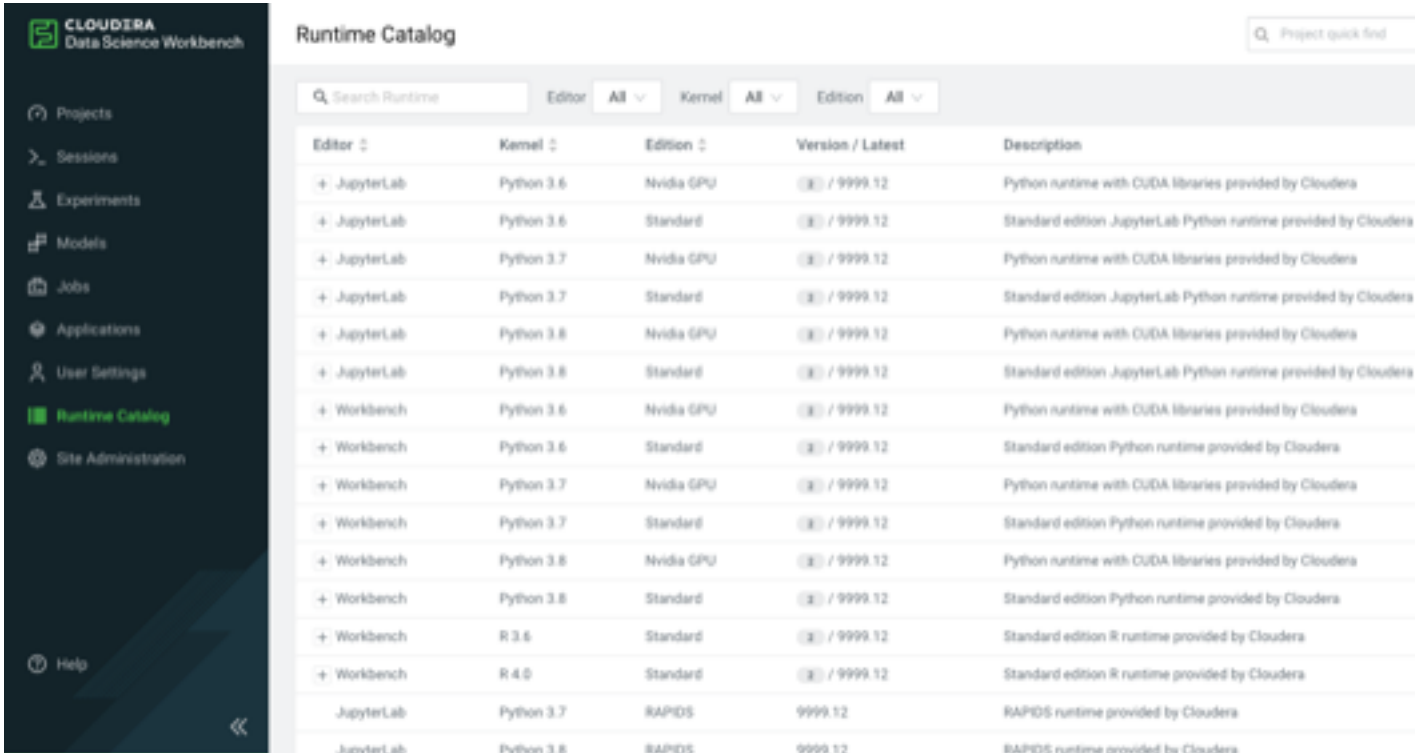
You can use the Runtime catalog to list all runtimes that are available for your deployment.

About this task

The Runtime Catalog lists information about each of the available runtimes. Information includes the editor and kernel supported by the runtime along with the edition, version, and a brief description.

Procedure

Click Runtime Catalog in the left Navigation bar to display a list of all available runtimes.



Editor	Kernel	Edition	Version / Latest	Description
+ JupyterLab	Python 3.6	Nvidia GPU	3 / 9999.12	Python runtime with CUDA libraries provided by Cloudera
+ JupyterLab	Python 3.6	Standard	3 / 9999.12	Standard edition JupyterLab Python runtime provided by Cloudera
+ JupyterLab	Python 3.7	Nvidia GPU	3 / 9999.12	Python runtime with CUDA libraries provided by Cloudera
+ JupyterLab	Python 3.7	Standard	3 / 9999.12	Standard edition JupyterLab Python runtime provided by Cloudera
+ JupyterLab	Python 3.8	Nvidia GPU	3 / 9999.12	Python runtime with CUDA libraries provided by Cloudera
+ JupyterLab	Python 3.8	Standard	3 / 9999.12	Standard edition JupyterLab Python runtime provided by Cloudera
+ Workbench	Python 3.6	Nvidia GPU	3 / 9999.12	Python runtime with CUDA libraries provided by Cloudera
+ Workbench	Python 3.6	Standard	3 / 9999.12	Standard edition Python runtime provided by Cloudera
+ Workbench	Python 3.7	Nvidia GPU	3 / 9999.12	Python runtime with CUDA libraries provided by Cloudera
+ Workbench	Python 3.7	Standard	3 / 9999.12	Standard edition Python runtime provided by Cloudera
+ Workbench	Python 3.8	Nvidia GPU	3 / 9999.12	Python runtime with CUDA libraries provided by Cloudera
+ Workbench	Python 3.8	Standard	3 / 9999.12	Standard edition Python runtime provided by Cloudera
+ Workbench	R 3.6	Standard	3 / 9999.12	Standard edition R runtime provided by Cloudera
+ Workbench	R 4.0	Standard	3 / 9999.12	Standard edition R runtime provided by Cloudera
JupyterLab	Python 3.7	RAPIDS	9999.12	RAPIDS runtime provided by Cloudera
JupyterLab	Python 3.8	RAPIDS	9999.12	RAPIDS runtime provided by Cloudera

Related Information

[Customized Runtimes](#)

ML Runtimes Nvidia GPU Edition

The NVIDIA GPU Edition Runtimes are built on top of NVIDIA CUDA docker images. CUDA is a parallel computing platform and programming model developed by NVIDIA for general computing on graphical processing units (GPUs). With CUDA, developers can dramatically speed up computing applications by harnessing the power of GPUs.

Cloudera Data Science Workbench supports GPUs using particular ML Runtime Editions. NVIDIA GPUs together with related specialized software can be utilized using the NVIDIA GPU Edition.



Note: If you are using Cloudera Data Science Workbench, to enable GPU usage make sure to follow the steps to provision the CDSW hosts. See [NVIDIA/Enabling Cloudera Data Science Workbench to use GPUs](#).

Version information:

Version	Additional software	Tested OS	Tested Driver Version
2021.09	CUDA 11.4	RHEL 7.6	NVIDIA driver 460.56

Compatibility information:

Runtime Version	CUDA Version	Kernels	Editors	Base Image
2021.02	CUDA 11.1	Python 3.7 and Python 3.8	Workbench and JupyterLab	

ML Runtimes inherit the compatibility requirements for NVIDIA CUDA. For compatibility information on additional software and required driver versions, visit the following pages:

- [NVIDIA/CUDA compatibility matrix](#)
- [Tensorflow requirements](#)
- [Pytorch requirements](#)

Testing ML Runtime GPU Setup

You can use the following simple examples to test whether the new ML Runtime is able to leverage GPUs as expected.

1. Go to a project that is using the ML Runtimes NVIDIA GPU edition and click Open Workbench.
2. Launch a new session with GPUs.
3. Run the following command in the workbench command prompt to verify that the driver was installed correctly:

```
! /usr/bin/nvidia-smi
```

4. Use any of the following code samples to confirm that the new engine works with common deep learning libraries.

Pytorch

```
!pip3 install torch==1.4.0
from torch import cuda
assert cuda.is_available()
assert cuda.device_count() > 0
print(cuda.get_device_name(cuda.current_device()))
```

Tensorflow

```
!pip3 install tensorflow-gpu==2.1.0
from tensorflow.python.client import device_lib
assert 'GPU' in str(device_lib.list_local_devices())
device_lib.list_local_devices()
```

Keras

```
!pip3 install keras
from keras import backend
assert len(backend.tensorflow_backend._get_available_gpus()) > 0
print(backend.tensorflow_backend._get_available_gpus())
```


ML Runtimes NVIDIA RAPIDS Edition

The RAPIDS Edition Runtimes are built on top of community built RAPIDS docker images. The RAPIDS suite of software libraries gives you the freedom to execute end-to-end data science and analytics pipelines entirely on GPUs. It relies on NVIDIA CUDA primitives for low-level compute optimization, but exposes that GPU parallelism and high-bandwidth memory speed through user-friendly Python interfaces.



Note: RAPIDS require NVIDIA Pascal or better GPUs. You need to use P3 or newer EC2 instances on AWS to meet this requirement.

Visit rapids.ai for more information.



Note: RAPIDS on Spark is not support as Spark3 is not supported on CDSW.

ML Runtimes RAPIDS edition differs from other Runtime Editions in the following ways:

- Python maintenance versions differ from what is being used in the Standard and NVIDIA GPU edition runtimes. These Python kernels are coming from the RAPIDS base image.
- Pre-installed Python packages and package versions differ from what's in Standard and NVIDIA GPU edition runtimes.

RAPIDS Runtimes are an optional extension of the Cloudera ML Runtime distribution. The RAPIDS images are not distributed automatically. Administrators can register them in the Runtime Catalog. The following RAPIDS editions are available for the 2021.04 Runtime version:

RAPIDS and CUDA Version	Kernel	Editor	Base Image	Docker
RAPIDS 0.18 CUDA 11.0	Python 3.7	Workbench editor	rapidsai/rapidsai-core:0.18-cuda11.0-base-ubuntu20.04-py3.7	docker.repository.cloudera.com/cloud-era/cdsw/ml-runtime-workbench-python3.7-rapids:2021.12.1-b17
RAPIDS 0.18 CUDA 11.0	Python 3.8	Workbench editor	rapidsai/rapidsai-core:0.18-cuda11.0-base-ubuntu20.04-py3.8	docker.repository.cloudera.com/cloud-era/cdsw/ml-runtime-workbench-python3.8-rapids:2021.12.1-b17
RAPIDS 0.18 CUDA 11.0	Python 3.7	JupyterLab editor	rapidsai/rapidsai-core:0.18-cuda11.0-base-ubuntu20.04-py3.7 Note: This image isn't based on NVIDIA's JupyterLab installation so RAPIDS library examples are not installed.	docker.repository.cloudera.com/cloud-era/cdsw/ml-runtime-jupyterlab-python3.7-rapids:2021.12.1-b17

RAPIDS and CUDA Version	Kernel	Editor	Base Image	Docker
RAPIDS 0.18 CUDA 11.0	Python 3.8	JupyterLab editor	rapidsai/rapidsai-core:0.18-cuda11.0-base-ubuntu20.04-py3.8 Note: This image isn't based on NVIDIA's JupyterLab installation so RAPIDS library examples are not installed.	docker.repository.cloudera.com/cloud-era/cds/ml-runtime-jupyterlab-python3.8-rapids:2021.12.1-b17

RAPIDS Python environment

The RAPIDS python libraries are distributed and present as a conda environment in the docker image. While it's possible to install additional libraries for your project via conda, these packages will be installed to a non-persistent part of the file system and won't persist between sessions and jobs.

Installing additional libraries via pip is supported and works the same way as in other Runtimes.



Note: the RAPIDS images have the official Anaconda channels configured for conda, if you decide to use conda to install packages, make sure you are familiar with Anaconda's license requirements.

Related Information

[Testing ML Runtime GPU Setup](#)

Using Editors for ML Runtimes

Cloudera Data Science Workbench provides a Workbench UI to edit and run code, but we also provide a preconfigured JupyterLab runtime to allow this. Choose the editor you prefer when launching a ML Runtime session.

Using JupyterLab with ML Runtimes

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data.

You can use JupyterLab to configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab allows you to use extensions that add new components and integrate with existing ones.

You must install any files necessary for JupyterLab, including configuration, customization, extensions, and kernels, into `/home/cds` for each project.

Code Completion in JupyterLab

You can use JupyterLab to write and run code in notebooks or in a traditional `.py` file. Completion works out of the box in notebooks and consoles. If you are working in a `.py` file in the text editor, you must create a console to provide the completion suggestions. To do this:

1. Open the `.py` file.
2. Right click on the document and choose "Create console for editor".
3. Choose the option appropriate to the type of completion you want to perform.

For example, to get completions on "math", you must first "import math" in the console.

Set JupyterLab Timeouts

You can control JupyterLab timeouts with the following environment variables:

```
JUPYTER_KERNEL_TIMEOUT_SECONDS  
JUPYTER_SERVER_TIMEOUT_SECONDS
```

Installing a Jupyter extension

Extensions modify the appearance or behaviour of Jupyter applications (including JupyterLab and Jupyter Notebook).

Before you begin

You must install any files necessary for JupyterLab, including configuration, customization, extensions, and kernels, into /home/cdsw for each project.



Note: Any Jupyter Extension that is not specifically designed for JupyterLab should work fine with Jupyter Notebook, which is preinstalled in Legacy Engine engine:14.

About this task

The following example installs the ipython-sql extension. This extension adds a %sql magic command that allows you to communicate directly with any database supported by SQLAlchemy.

Procedure

1. Launch a JupyterLab session.
2. Open a terminal and run the following command:

```
pip install ipython-sql
```



Note: The previous command is only an example for installing ipython-sql.

3. Open a notebook or console and run the following command:

```
%load_ext sql
```

Results

You can now use the %sql magic command to connect to a particular database, and the %%sql magic command to create a cell containing an SQL query to run against that database. See the JupyterLab documentation for additional information on Jupyter magic commands.

Installing a Jupyter kernel

Jupyter kernels are “connections” through which a notebook (or other part of JupyterLab) can talk to a particular interpreter. CDSW includes one kernel in each JupyterLab Runtime: Python 3.6, Python 3.7, or Python 3.8.

Jupyter kernels are “connections” through which a notebook (or other part of JupyterLab) can talk to an interpreter. ML Runtimes come with a Jupyter kernel for Python 3.6, 3.7 or 3.8 preinstalled, but you may also install Jupyter kernels for bash or JavaScript/TypeScript.

bash

The following example describes how to enable a bash kernel in a particular project.

1. Launch a JupyterLab session.

2. Open a terminal window and run the following command:

```
pip install bash_kernel
python -m bash_kernel.install
```

You should now have the option to launch bash notebooks and consoles.

JavaScript and TypeScript

The following example describes how to enable JavaScript and TypeScript kernels in a particular project. .

1. Customize your PATH environment variable:
 - a. Navigate to Project Settings/Advanced.
 - b. Set PATH to \$HOME/node_modules/.bin:\$PATH.
2. Launch a JupyterLab session.
3. Open a terminal window and run the following command:

```
npm install tslab
tslab install
```

You should now have the option to launch JavaScript and TypeScript notebooks and console.

Using Conda Runtime

Users now can create their own Python or R Conda environments within their CML Projects that they can use in the JupyterLab editor.

Environments, installed packages, and kernel specifications are persisted on the Project file system. An example flow for creating a new Python 3.10 Conda environment is the following:

```
conda create --name myenv python=3.10
conda activate myenv
conda install ipykernel
ipython kernel install --user --name=myenv
```

JupyterLab Conda Tech Preview Runtime

You might run into some known issues while using JupyterLab Conda Runtime.

Sessions

When starting a Notebook or a Console for a specific environment, the installed packages will be available and the interpreter used to evaluate the contents of the Notebook or Console will be the one installed in the environment. However, the Conda environment is not "activated" in these sessions, therefore commands like `!which python` will return with the base Python 3.10 interpreter on the Runtime. The recommended ways to modify a Conda environments or install packages are the following:

- conda commands must be used with the `-n` or `--name` argument to specify the environment, for example `conda -n myenv install pandas`
- When installing packages with pip, use the `%pip` magic to install packages in the active kernel's environment, for example `%pip install pandas`

Applications and Jobs

To start an Application or Job, first create a launcher Python script containing the following line: `!source activate <conda_env_name> && python <job / application script.py>`

When starting the Application or Job, select the launcher script as the "Script".

Models

Models are currently not supported for the Conda Runtime.

Spark

Spark is not supported in JupyterLab Notebooks and Consoles.

Spark workloads are supported in activated Conda environments in JupyterLab Terminals, or in Jobs or Applications.

The CDSW libraries for Python and R are not available for the Conda Runtimes.

Installing Additional ML Runtimes Packages

ML Runtimes are preloaded with a few common packages and libraries for Python. However, a key feature of Cloudera Data Science Workbench is the ability of different projects to install and use libraries pinned to specific versions, just as you would on your local computer.

Generally, Cloudera recommends you install all required packages locally into your project. This will ensure you have the exact versions you want and that these libraries will not be upgraded when Cloudera upgrades the ML Runtimes image. You only need to install libraries and packages once per project. From then on, they are available to any new ML Runtimes you spawn throughout the lifetime of the project.

You can install additional libraries and packages from the workbench, using either the command prompt or the terminal.



Note:

Cloudera Data Science Workbench does not currently support installation of packages that require root access to the hosts. For such use-cases, you will need to create a new custom engine that extends the base engine image to include the required packages. For instructions, see [Customized Runtimes](#) on page 17.

(Python) Install Packages Using Workbench Command Prompt

To install a package from the command prompt:

1. Navigate to your project's Overview page. Click Open Workbench and launch a session.
2. At the command prompt (see Native Workbench Console and Editor) in the bottom right, enter the command to install the package. Some examples using Python have been provided.

Python 3

```
# Installing from console using ! shell operator and pip3:  
!pip3 install beautifulsoup4  
# Installing from terminal  
pip3 install beautifulsoup4
```

(Python Only) Using a Requirements File

For a Python project, you can specify a list of the packages you want in a requirements.txt file that lives in your project. The packages can be installed all at once using pip/pip3.

1. Create a new file called requirements.txt file within your project:

```
beautifulsoup4==4.6.0  
seaborn==0.7.1
```

2. To install the packages in a Python 3 ML Runtimes, run the following command in the workbench command prompt.

```
!pip3 install -r requirements.txt
```

Restrictions for upgrading R and Python packages

Some R and Python packages should not be upgraded because doing so will break the Workbench UI.

For R Runtimes

Do not upgrade the Cairo or RServe packages in your projects using R Runtimes. If you do, you may be unable to launch sessions, jobs, experiments, or applications.

If these packages are upgraded, you will need to start a new project or to delete the directory `.local/lib` using the CDSW/CML Files UI ("show hidden").

For Python Runtimes

Do not upgrade the ipykernel package or its dependencies (ipython, traitlets, jupyter_client, and tornado) in your project using Python Runtimes. If you do, you may be unable to launch sessions, jobs, experiments, or applications.

If these packages are upgraded, you will need to start a new project or to delete the directory `.local/lib` using the CDSW/CML Files UI ("show hidden").

Python Runtimes in CML fail to import the setuptools Python library and can fail installing some Python packages

Python Runtimes in CML fail to import the setuptools Python library and therefore can fail installing some Python packages when the library setuptools is present on the Runtime or is installed into the CML project with version 60.0.0 or higher.

Python 3.10 Runtimes from the 2023.05 Runtime release ship with a newer version of setuptools, so customers can run into this issue when they are using that Runtime. Also they can run into this issue when they are using Custom Runtimes that has a newer setuptools library version or when they install a new setuptools version into their project (regardless of what Runtime they use).

Workaround: Set the environmental variable `SETUPTOOLS_USE_DISTUTILS=stdlib` either on a project level under Project Settings -> Advanced or on a workspace level under Site Administration -> Runtime -> Environment variables.

ML Runtimes Environment Variables

This topic describes how ML Runtimes environmental variables work. It also lists the different scopes at which they can be set and the order of precedence that will be followed in case of conflicts.

ML Runtimes environment variables behave the same way for Legacy Engines and ML Runtimes.

Environmental variables allow you to customize ML Runtimes environments for projects. For example, if you need to configure a particular timezone for a project, or increase the length of the session/job timeout windows, you can use environmental variables to do so. Environmental variables can also be used to assign variable names to secrets such as passwords or authentication tokens to avoid including these directly in the code.

In general, Cloudera recommends that you do not include passwords, tokens, or any other secrets directly in your code because anyone with read access to your project will be able to view this information. A better place to store secrets is in your project's environment variables, where only project collaborators and admins have view access. They can therefore be used to securely store confidential information such as your AWS keys or database credentials.

Cloudera Data Science Workbench allows you to define environmental variables for the following scopes:

Global

A site administrator for your Cloudera Data Science Workbench deployment can set environmental variables on a global level. These values will apply to every project on the deployment.

To set global environmental variables, go to Admin Runtime/Engine .

Project

Project administrators can set project-specific environmental variables to customize the ML Runtimes launched for a project. Variables set here will override the global values set in the site administration panel.

To set environmental variables for a project, go to the project's Overview page and click **Settings Advanced**.

Job

Environments for individual jobs within a project can be customized while creating the job. Variables set per-job will override the project-level and global settings.

To set environmental variables for a job, go to the job's Overview page and click **Settings Set Environmental Variables**.

Experiments

ML Runtimes created for execution of experiments are completely isolated from the project. However, these ML Runtimes inherit values from environmental variables set at the project-level and/or global level. Variables set at the project-level will override the global values set in the site administration panel.

Models


Model environments are completely isolated from the project. Environmental variables for these ML Runtimes can be configured during the build stage of the model deployment process. Models will also inherit any environment variables set at the project and global level. However, variables set per-model build will override other settings.

ML Runtimes Environment Variables List

The following table lists Cloudera Data Science Workbench environment variables that you can use to customize your project environments. These can be set either as a site administrator or within the scope of a project or a job.

Environment Variable	Description
MAX_TEXT_LENGTH	Maximum number of characters that can be displayed in a single text cell. By default, this value is set to 800,000 and any more characters will be truncated. Default: 800,000
SESSION_MAXIMUM_MINUTES	Maximum number of minutes a session can run before it times out. Default: 60*24*7 minutes (7 days) Maximum Value: 35,000 minutes
JOB_MAXIMUM_MINUTES	Maximum number of minutes a job can run before it times out. Default: 60*24*7 minutes (7 days) Maximum Value: 35,000 minutes
IDLE_MAXIMUM_MINUTES	Maximum number of minutes a session can remain idle before it exits. Default: 60 minutes Maximum Value: 35,000 minutes

Per-Engine Environmental Variables: In addition to the previous table, there are some more built-in environmental variables that are set by the Cloudera Data Science Workbench application itself and do not need to be modified by users. These variables are set per-engine launched by Cloudera Data Science Workbench and only apply within the scope of each engine.

Environment Variable	Description
CDSW_PROJECT	The project to which this engine belongs.
CDSW_ENGINE_ID	The ID of this engine. For sessions, this appears in your browser's URL bar.
CDSW_MASTER_ID	If this engine is a worker, this is the CDSW_ENGINE_ID of its master.
CDSW_MASTER_IP	If this engine is a worker, this is the IP address of its master.
CDSW_PUBLIC_PORT	 Note: This property is deprecated. See CDSW_APP_PORT and CDSW_READONLY_PORT for alternatives. A port on which you can expose HTTP services in the engine to browsers. HTTP services that bind CDSW_PUBLIC_PORT will be available in browsers at: <code>http(s)://<CDSW_ENGINE_ID>.<CDSW_DOMAIN></code> . By default, CDSW_PUBLIC_PORT is set to 8080. A direct link to these web services will be available from the grid icon in the upper right corner of the Cloudera Data Science Workbench web application, as long as the job or session is still running. For more details, see <i>Accessing Web User Interfaces from Cloudera Data Science Workbench</i> . In Cloudera Data Science Workbench, setting CDSW_PUBLIC_PORT to a non-default port number is not supported.
CDSW_APP_PORT	A port on which you can expose HTTP services in the engine to browsers. HTTP services that bind CDSW_APP_PORT will be available in browsers at: <code>http(s)://<CDSW_ENGINE_ID>.<CDSW_DOMAIN></code> . Use this port for applications that grant some control to the project, such as access to the session or terminal. A direct link to these web services will be available from the grid icon in the upper right corner of the Cloudera Machine Learning web application as long as the job or session runs. Even if the web UI does not have authentication, only Contributors and those with more access to the project can access it. For more details, see <i>Accessing Web User Interfaces from Cloudera Data Science Workbench</i> . Note that if the Site Administrator has enabled Allow only session creators to run commands on active sessions, then the UI is only available to the session creator. Other users will not be able to access it. Use 127.0.0.1 as the IP.
CDSW_READONLY_PORT	A port on which you can expose HTTP services in the engine to browsers. HTTP services that bind CDSW_READONLY_PORT will be available in browsers at: <code>http(s)://<CDSW_ENGINE_ID>.<CDSW_DOMAIN></code> . Use this port for applications that grant read-only access to project results. A direct link to these web services will be available to users with from the grid icon in the upper right corner of the Cloudera Data Science Workbench web application as long as the job or session runs. Even if the web UI does not have authentication, Viewers and those with more access to the project can access it. For more details, see <i>Accessing Web User Interfaces from Cloudera Data Science Workbench</i> . Use 127.0.0.1 as the IP.
CDSW_DOMAIN	The domain on which Cloudera Data Science Workbench is being served. This can be useful for iframing services, as demonstrated in <i>Accessing Web User Interfaces from Cloudera Data Science Workbench</i> .
CDSW_CPU_MILLICORES	The number of CPU cores allocated to this engine, expressed in thousandths of a core.
CDSW_MEMORY_MB	The number of megabytes of memory allocated to this engine.
CDSW_IP_ADDRESS	Other engines in the Cloudera Data Science Workbench cluster can contact this engine on this IP address.

Accessing Environmental Variables from Projects

This topic shows you how to access environmental variables from your code.

Environmental variables are injected into every engine launched for a project, contingent on the scope at which the variable was set (global, project, etc.). The following code samples show how to access a sample environment variable called `DATABASE_PASSWORD` from your project code.

Python

```
import os
database_password = os.environ["DATABASE_PASSWORD"]
```

Appending Values to Environment Variables:

You can also set environment variables to append to existing values instead of replacing them. For example, when setting the `LD_LIBRARY_PATH` variable, you can set the value to `LD_LIBRARY_PATH:/path/to/set`.

Customized Runtimes

This topic explains how custom Runtimes work and when they should be used.



Note: Cloudera, Inc. (“Cloudera”) makes the Custom Runtime feature available to allow its users to add, run, and manage their own container images for their workloads as ‘Custom Runtimes’. Cloudera’s support covers only the integration points of these images with Cloudera Machine Learning, to the extent adding, running, and managing Custom Runtime images fulfill the documented requirements and/or pre-requisites. Cloudera does not provide any support for any other configurations and/or third party software added or installed to the image by customers.

By default, Cloudera Data Science Workbench Runtimes are preloaded with a few common packages and libraries for R, Python, and Scala. In addition to these, Cloudera Data Science Workbench also allows you to install any other packages or libraries that are required by your projects. However, directly installing a package to a project as described above might not always be feasible. For example, packages that require root access to be installed, or that must be installed to a path outside `/home/cdsw` (outside the project mount), cannot be installed directly from the workbench.

For such circumstances, Cloudera Data Science Workbench allows you to extend the base Docker image and create a new Docker image with all the libraries and packages you require. Site administrators can then add this new image in the allowlist for use in projects.



Note: You will need to remove any unnecessary Cloudera sources or repositories that are inaccessible because of the payroll.

Note that this approach can also be used to accelerate project setup across the deployment. For example, if you want multiple projects on your deployment to have access to some common dependencies (package or software or driver) out of the box, or even if a package just has a complicated setup, it might be easier to simply provide users with a Runtime that has already been customized for their project(s).

Related Resources

- The Cloudera Engineering Blog post on *Customizing Docker Images in Cloudera Data Science Workbench* describes an end-to-end example on how to build and publish a customized Docker image and use it as an engine in Cloudera Data Science Workbench.
- For an example of how to extend the base engine image to include Conda, see *Installing Additional Packages*.

Creating Customized ML Runtimes

This section walks you through the steps required to create your own custom ML Runtimes based on one of the Cloudera provided ML Runtime images.

Create a Dockerfile for the Custom Runtime Image

This topic shows you how to create a Dockerfile for a custom image.

First, select an appropriate source image for your customization. For a non-PBJ Runtime, you must use a Runtime image released by Cloudera. Image tags can be seen on the Session Start page on the user interface when you select a Runtime. The second step when building a customized image is to create a Dockerfile that specifies which packages you would like to install in addition to the base image.

For example, the following Dockerfile installs the telnet package, the sklearn Python package and upgraded base packages on top of an ML Runtime image released by Cloudera.

```
# Dockerfile
# Specify an ML Runtime base image
FROM docker.repository.cloudera.com/cloudera/cds/ml-runtime-jupyterlab-pyth
on3.7-standard:2021.12.1-b17
# Install telnet in the new image
RUN apt-get update && apt-get install -y --no-install-recommends telnet &&
  apt-get clean && rm -rf /var/lib/apt/lists/*
# Upgrade packages in the base image
RUN apt-get update && apt-get upgrade -y && apt-get clean && rm -rf /var/lib
/apt/lists/*
# Install the python package sklearn
RUN pip install --no-cache-dir sklearn
# Override Runtime label and environment variables metadata
ENV ML_RUNTIME_EDITION="Telnet Edition" \
    ML_RUNTIME_SHORT_VERSION="1.0" \
    ML_RUNTIME_MAINTENANCE_VERSION=1 \
    ML_RUNTIME_DESCRIPTION="This runtime includes telnet and sklearn
and upgraded packages"
ENV ML_RUNTIME_FULL_VERSION="${ML_RUNTIME_SHORT_VERSION}.${ML_RUNTIME_MAI
NTENANCE_VERSION}"
LABEL com.cloudera.ml.runtime.edition=$ML_RUNTIME_EDITION \
    com.cloudera.ml.runtime.full-version=$ML_RUNTIME_FULL_VERSION \
    com.cloudera.ml.runtime.short-version=$ML_RUNTIME_SHORT_VERSION \
    com.cloudera.ml.runtime.maintenance-version=$ML_RUNTIME_MAINTENANCE
_VERSION \
    com.cloudera.ml.runtime.description=$ML_RUNTIME_DESCRIPTION
```

Metadata for Custom ML Runtimes

This topic addresses the metadata for custom Runtimes.

All new custom Runtimes must override the Edition metadata of existing Runtimes. The rest of the metadata can be overridden to communicate the expectations for the consumers of the image. Both the Docker label and the environment variable match in a custom Runtime image. In order to add or register a custom Runtime to a deployment, the user facing metadata combination should be unique in that deployment. For example, of the following, Editor, Edition, Kernel, Version, and Maintenance Version, at least the later should be incremented for adding a next iteration of the same image.

See the following reference table for more details on ML Runtime metadata.

Environment variable	Docker Label	Description	Override in custom non-PBJ runtime
ML_RUNTIME_METADATA_VERSION	com.cloudera.ml.runtime.runtime-metadata-version	Metadata version	Not allowed
ML_RUNTIME_EDITOR	com.cloudera.ml.runtime.editor	CDSW/CML Editor installed in the image.	Allowed
ML_RUNTIME_EDITION	com.cloudera.ml.runtime.edition	Edition of the image, a notion of the Runtime capabilities.	Required

ML_RUNTIME_DESCRIPTION	com.cloudera.ml.runtime.description	Longer description of the Runtime image capabilities.	Recommended
ML_RUNTIME_KERNEL	com.cloudera.ml.runtime.kernel	Main kernel included in the image, e.g., Python 3.8	Not allowed
ML_RUNTIME_SHORT_VERSION	com.cloudera.ml.runtime.short-version	Main version of the image, e.g., 1.0. This shows up as Version in the selection screen.	Recommended
ML_RUNTIME_FULL_VERSION	com.cloudera.ml.runtime.full-version	Full version consists of the short version + maintenance version, e.g., 1.0.1	Optional
ML_RUNTIME_MAINTENANCE_VERSION	com.cloudera.ml.runtime.maintenance-version	Maintenance version must be an integer, e.g., 1. Only the largest maintenance version of the set of the same short version images are visible for users to select. Increment this number if you create a drop in replacement of an existing Runtime. Start with 1 with a new version or edition.	Recommended
ML_RUNTIME_CUDA_VERSION	com.cloudera.ml.runtime.cuda-version	CUDA version installed in the image.	Not allowed
ML_RUNTIME_GIT_HASH	com.cloudera.ml.runtime.git-hash	Git hash of runtime source	Not allowed
ML_RUNTIME_GBN	com.cloudera.ml.runtime.gbn	Cloudera internal build number	Not allowed

Editor Customization

This topic addresses customizing a third-party editor to work with ML Runtimes.

For a third-party editor to work with ML Runtimes, you must provide a starting script. For example:

```
#!/bin/bash
"$ZEPPELIN_HOME/bin/zeppelin.sh"
```

For Cloudera Data Science Workbench to interpret this as an editor startup script, you must create a symlink to the editor as `/usr/local/bin/ml-runtime-editor`. This will be created in the Dockerfile of the customized runtime.



Note: Third-party editors provide a way to run arbitrary code that is not distributed by Cloudera. Use third-party editors at your own risk. You should absolutely trust the code that you want to run.

Build the New Docker Image

This topic shows you how to use Docker to build a custom image.

A new custom Docker image can be built on any host where Docker binaries are installed, source images are available, and OS repositories and other package repositories are available. To install these binaries, run the following command on the host where you want to build the new image:

```
docker build -t <image-name>:<tag> . -f Dockerfile
```

If your Dockerfile makes any outside connection (for example, `apt-get`, `update`, `pip install`, `curl`), you must add the `--network=host` option to the build command:

```
docker build --network=host -t <image-name>:<tag> . -f Dockerfile
```

Distribute the Image

This topic explains the different methods that can be used to distribute a custom ML Runtime to all the hosts.

Once you have built a new custom ML Runtime, use one of the following methods to distribute the new image to all your Cloudera Data Science Workbench hosts:

Push the image to a public registry such as DockerHub

For instructions, refer to the Docker documentation *docker push* and *Push images to Docker Cloud*.

Push the image to your company's Docker registry

When using this method, make sure to tag your image with the following schema:

```
docker tag <image-name> <company-registry>/<user-name>/<image-name>:<tag>
```

Once the image has been tagged properly, use the following command to push the image:

```
docker push <company-registry>/<user-name>/<image-name>:<tag>
```

Add Docker registry credentials

To enable Cloudera Data Science Workbench to fetch custom ML Runtimes from a secure repository, as Administrator you need to add Docker registry credentials.

Create a `kubectl` secret named `regcred` for your secured Docker registry. The following command creates the secret in your Kubernetes cluster:

```
kubectl create secret docker-registry regcred
--docker-server=<server host>
--docker-username=<username>
--docker-password=<password>
-n <compute namespace eg. mlx>
```

The next time the ML Runtime image is pulled, the new secret will be picked up.



Note: It may help to automatically pick up the password, like so:

```
kubectl create secret docker-registry regcred
--docker-server=<server host>
--docker-username=AWS
--docker-password=$(aws ecr get-login-password --region <region>)
-n <compute namespace eg. mlx>
```

Add the new ML Runtime

Cloudera Data Science Workbench enables you to add customized ML Runtimes from the Runtime Catalog window.

About this task



Note: You must have system administrator permission to add a new ML Runtime.



Note: If you add a Custom Runtime from a private docker registry, you need to add the docker credentials first to CML. See *Add Docker registry credentials and certificates* for more information.

Procedure

1. Click Runtime Catalog from the Navigation panel.
2. Click the Add Runtime button in the upper right corner.

3. In the Add Runtime window, enter the url of the Runtime Docker image you want to upload.

As ML Runtimes are identified based on certain attributes, metadata (such as Editor, Kernel, Edition, Version, and Maintenance Version) must be unique to add new Customized Runtimes to a deployment. Customized ML Runtimes must have different Edition text compared to Cloudera supported versions.

4. Click Validate.

CML will use the provided URL to fetch the Docker image and validate if it can be used as a customized Runtime.

If the Docker image is successfully validated, CML will display the metadata information of the image. The new customized Runtime will be visible in the Runtime Catalog and accessible over the different workloads.

Limitations

This topic lists some limitations associated with customized ML Runtime images.

- The contents of certain pre-existing standard directories such as /home/cdsw, /tmp, and so on, cannot be modified while creating customized non-PBJ ML Runtimes. This means any files saved in these directories will not be accessible from sessions that are running on customized ML Runtimes.

Workaround: Create a new custom directory in the Dockerfile used to create the customized ML Runtime, and save your files to that directory.

For PBJ Runtimes, note the following limitations:

- PBJ Runtimes work as models only with R and Python kernels.

Pre-Installed Packages in ML Runtimes

Cloudera Data Science Workbench ships with several base engine images that include Python kernels, and frequently used libraries.



Note: New ML Runtime releases are automatically added to the deployment, if internet connection is available.

Because the software and versions in these lists is installed in the images, you should not upgrade these versions because this will cause dependency conflicts. If any software in this list needs to be at a higher version, you should use a more recent version of the legacy engine or ML Runtime.

ML Runtimes 2023.05

This section lists the Python, R, and Scala libraries that ship with ML Runtimes 2023.05.



Note: Nvidia GPU Editions with a Workbench Editor come with CUDA 11.4.1 preinstalled.

All other Nvidia GPU Edition Runtimes come with CUDA 11.8.0 preinstalled.

Python 3.10 Libraries for Conda

This section lists the packages in Python 3.10 Conda Runtimes that ship with ML Runtimes 2023.05.

Table 1: Packages in Python 3.10 Conda Runtimes - Standard

Library	Version
python	3.10.9

Library	Version
pip	23.1.2
aiofiles	22.1.0
aiosqlite	0.19.0
anyio	3.6.2
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
arrow	1.2.3
asttokens	2.2.1
attrs	23.1.0
Babel	2.12.1
backcall	0.2.0
beautifulsoup4	4.11.2
bitarray	2.7.3
bleach	6.0.0
cattrs	22.2.0
certifi	2022.12.7
cffi	1.15.1
charset-normalizer	3.1.0
colorama	0.4.6
comm	0.1.3
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2
defusedxml	0.7.1
docstring-to-markdown	0.12
entrypoints	0.4
exceptiongroup	1.1.1
executing	1.2.0
fastjsonschema	2.16.3
fonttools	4.39.3
fqdn	1.5.1
gitdb	4.0.10
GitPython	3.1.31
idna	3.4
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
ipython-genutils	0.2.0

Library	Version
isoduration	20.11.0
jedi	0.18.2
jedi-language-server	0.40.0
Jinja2	3.1.2
json5	0.9.11
jsonpointer	2.3
jjsonschema	4.17.3
jupyter-client	7.2.2
jupyter-events	0.6.3
jupyter-lsp	2.1.0
jupyter-server	1.23.4
jupyter-server-mathjax	0.2.6
jupyter-ydoc	0.2.4
jupyter_core	5.3.0
jupyter_server_fileid	0.9.0
jupyter_server_ydoc	0.6.1
jupyterlab	3.6.0
jupyterlab-git	0.41.0
jupyterlab-lsp	4.1.0
jupyterlab-pygments	0.2.2
jupyterlab_server	2.20.0
kerberos	1.3.1
kiwisolver	1.4.4
lsprotocol	2023.0.0a1
MarkupSafe	2.1.2
matplotlib	3.7.0
matplotlib-inline	0.1.3
mistune	2.0.5
nbcassic	0.5.3
nbclient	0.6.3
nbconvert	7.2.9
nbdime	3.2.0
nbformat	5.7.3
nest-asyncio	1.5.5
notebook	6.5.4
notebook_shim	0.2.2
numpy	1.24.2
packaging	23.1
pandocfilters	1.5.0

Library	Version
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.5.0
platformdirs	3.5.0
prometheus-client	0.16.0
prompt-toolkit	3.0.8
psutil	5.9.5
ptyprocess	0.6.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.21
pydantic	1.10.7
pygls	1.0.1
Pygments	2.8.1
pyparsing	3.0.9
pyrsistent	0.19.3
python-dateutil	2.8.2
python-json-logger	2.0.7
PyYAML	6.0
pyzmq	22.3.0
requests	2.28.2
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1
Send2Trash	1.8.0
six	1.16.0
smmap	5.0.0
sniffio	1.3.0
soupsieve	2.4
stack-data	0.6.2
terminado	0.17.1
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.2.1
tomli	2.0.1
tornado	6.1
traitlets	5.9.0
typeguard	2.13.3

Library	Version
typing_extensions	4.5.0
uri-template	1.2.0
urllib3	1.26.14
wcwidth	0.2.5
webcolors	1.13
webencodings	0.5.1
websocket-client	1.5.1
y-py	0.5.9
ypy-websocket	0.8.2

Python 3.10 Libraries for Workbench

This section lists the packages in Python 3.10 Workbench Runtimes that ship with ML Runtimes 2023.05.

Table 2: Packages in Python 3.10.9 PBJ Workbench Runtimes - Standard

Library	Version
python	3.10.9
pip	23.1.2
anyio	3.6.2
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
arrow	1.2.3
asttokens	2.2.1
attrs	22.2.0
backcall	0.2.0
beautifulsoup4	4.12.0
bitarray	2.7.3
bleach	6.0.0
certifi	2022.12.7
cffi	1.15.1
charset-normalizer	3.1.0
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.6
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
executing	1.2.0
fastjsonschema	2.16.3
fonttools	4.39.3

Library	Version
fqdn	1.5.1
idna	3.4
impyla	0.18.0
ipykernel	6.9.0
ipython	8.1.0
ipython-genutils	0.2.0
isoduration	20.11.0
jedi	0.18.2
Jinja2	3.1.2
jsonpointer	2.3
jsonschema	4.17.3
jupyter-events	0.6.3
jupyter-kernel-gateway	2.5.1
jupyter_client	7.4.9
jupyter_core	5.3.0
jupyter_server	2.5.0
jupyter_server_terminals	0.4.4
jupyterlab-pygments	0.2.2
kerberos	1.3.1
kiwisolver	1.4.4
MarkupSafe	2.1.2
matplotlib	3.7.0
matplotlib-inline	0.1.6
mistune	2.0.5
nbclassic	0.5.5
nbclient	0.7.3
nbconvert	7.3.0
nbformat	5.8.0
nest-asyncio	1.5.6
notebook	6.5.3
notebook_shim	0.2.2
numpy	1.24.2
packaging	23.1
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.5.0
platformdirs	3.2.0

Library	Version
prometheus-client	0.16.0
prompt-toolkit	3.0.38
ptyprocess	0.7.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.21
Pygments	2.14.0
pyparsing	3.0.9
pyrsistent	0.19.3
python-dateutil	2.8.2
python-json-logger	2.0.7
PyYAML	6.0
pyzmq	25.0.2
requests	2.28.2
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1
Send2Trash	1.8.0
six	1.16.0
sniffio	1.3.0
soupsieve	2.4
stack-data	0.6.2
terminado	0.17.1
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.2.1
tornado	6.2
traitlets	5.9.0
uri-template	1.2.0
urllib3	1.26.15
wcwidth	0.2.6
webcolors	1.13
webencodings	0.5.1
websocket-client	1.5.1

Table 3: Packages in Python 3.10.9 PBJ Workbench Runtimes - Nvidia GPU Edition

Library	Version
python	3.10.9
pip	23.1.2

Library	Version
anyio	3.6.2
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
arrow	1.2.3
asttokens	2.2.1
attrs	22.2.0
backcall	0.2.0
beautifulsoup4	4.12.0
bitarray	2.7.3
bleach	6.0.0
certifi	2022.12.7
cffi	1.15.1
charset-normalizer	3.1.0
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.6
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
executing	1.2.0
fastjsonschema	2.16.3
fonttools	4.39.3
fqdn	1.5.1
idna	3.4
impyla	0.18.0
ipykernel	6.9.0
ipython	8.1.0
ipython-genutils	0.2.0
isoduration	20.11.0
jedi	0.18.2
Jinja2	3.1.2
jsonpointer	2.3
jsonschema	4.17.3
jupyter-events	0.6.3
jupyter-kernel-gateway	2.5.1
jupyter_client	7.4.9
jupyter_core	5.3.0
jupyter_server	2.5.0
jupyter_server_terminals	0.4.4

Library	Version
jupyterlab-pygments	0.2.2
kerberos	1.3.1
kiwisolver	1.4.4
MarkupSafe	2.1.2
matplotlib	3.7.0
matplotlib-inline	0.1.6
mistune	2.0.5
nbclassic	0.5.5
nbclient	0.7.3
nbconvert	7.3.0
nbformat	5.8.0
nest-asyncio	1.5.6
notebook	6.5.3
notebook_shim	0.2.2
numpy	1.24.2
packaging	23.1
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.5.0
platformdirs	3.2.0
prometheus-client	0.16.0
prompt-toolkit	3.0.38
ptyprocess	0.7.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.21
Pygments	2.14.0
pyparsing	3.0.9
pyrsistent	0.19.3
python-dateutil	2.8.2
python-json-logger	2.0.7
PyYAML	6.0
pyzmq	25.0.2
requests	2.28.2
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1

Library	Version
Send2Trash	1.8.0
six	1.16.0
sniffio	1.3.0
soupsieve	2.4
stack-data	0.6.2
terminado	0.17.1
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.2.1
tornado	6.2
traitlets	5.9.0
uri-template	1.2.0
urllib3	1.26.15
wcwidth	0.2.6
webcolors	1.13
webencodings	0.5.1
websocket-client	1.5.1

Python 3.9 Libraries for Workbench

This section lists the packages in Python 3.9 Workbench Runtimes that ship with ML Runtimes 2023.05.

Table 4: Packages in Python 3.9.11 Workbench Runtimes - Standard

Library	Version
python	3.9.11
pip	21.1.3
argcomplete	2.0.0
asttokens	2.2.1
backcall	0.2.0
bitarray	2.6.0
certifi	2021.5.30
charset-normalizer	2.0.6
comm	0.1.3
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2
entrypoints	0.4
executing	1.2.0
fonttools	4.39.3

Library	Version
idna	3.2
importlib-metadata	4.11.3
importlib-resources	5.12.0
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
jedi	0.18.2
jupyter-client	7.2.2
jupyter_core	5.3.0
kerberos	1.3.1
kiwisolver	1.4.4
matplotlib	3.7.0
matplotlib-inline	0.1.3
nest-asyncio	1.5.5
numpy	1.21.6
packaging	23.1
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
platformdirs	3.5.0
prompt-toolkit	3.0.8
psutil	5.9.5
ptyprocess	0.6.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	3.0.9
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.26.0
six	1.16.0
stack-data	0.6.2
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.9.0
typing_extensions	4.3.0

Library	Version
urllib3	1.26.6
wcwidth	0.2.5
zipp	3.15.0

Table 5: Packages in Python 3.9.11 Workbench Runtimes - Nvidia GPU Edition

Library	Version
python	3.9.11
pip	21.1.3
cuda	11.4.1-1
libcudnn8	8.2.2.26-1
argcomplete	2.0.0
asttokens	2.2.1
backcall	0.2.0
bitarray	2.6.0
certifi	2021.5.30
charset-normalizer	2.0.6
comm	0.1.3
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2
entrypoints	0.4
executing	1.2.0
fonttools	4.39.3
idna	3.2
importlib-metadata	4.11.3
importlib-resources	5.12.0
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
jedi	0.18.2
jupyter-client	7.2.2
jupyter_core	5.3.0
kerberos	1.3.1
kiwisolver	1.4.4
matplotlib	3.7.0
matplotlib-inline	0.1.3
nest-asyncio	1.5.5
numpy	1.21.6

Library	Version
packaging	23.1
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
platformdirs	3.5.0
prompt-toolkit	3.0.8
psutil	5.9.5
ptyprocess	0.6.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	3.0.9
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.26.0
six	1.16.0
stack-data	0.6.2
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.9.0
typing_extensions	4.3.0
urllib3	1.26.6
wcwidth	0.2.5
zipp	3.15.0

Table 6: Packages in Python 3.9.11 PBJ Workbench Runtimes - Standard

Library	Version
python	3.9.6
pip	21.1.3
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
asttokens	2.2.1
attrs	21.4.0
backcall	0.2.0
bitarray	2.6.0
bleach	4.1.0

Library	Version
certifi	2021.10.8
cff	1.15.0
charset-normalizer	2.0.11
contourpy	1.0.7
cycler	0.11.0
debugpy	1.5.1
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
executing	0.4
fonttools	4.39.3
idna	3.3
importlib-metadata	4.11.0
importlib-resources	5.4.0
impyla	0.18.0
ipykernel	6.9.0
ipython	8.1.0
ipython-genutils	0.2.0
jedi	0.18.1
Jinja2	3.0.3
jsonschema	4.4.0
jupyter-client	7.1.2
jupyter-core	4.11.2
jupyter-kernel-gateway	2.5.1
jupyterlab-pygments	0.1.2
kerberos	1.3.1
kiwisolver	1.4.4
MarkupSafe	2.0.1
matplotlib	3.7.0
matplotlib-inline	0.1.3
mistune	0.8.4
nbclient	0.5.10
nbconvert	6.4.2
nbformat	5.1.3
nest-asyncio	1.5.4
notebook	6.4.12
numpy	1.21.6
packaging	21.3
pandocfilters	1.5.0

Library	Version
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
prometheus-client	0.13.1
prompt-toolkit	3.0.27
ptyprocess	0.7.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.21
Pygments	2.11.2
pyparsing	3.0.9
pyrsistent	0.18.1
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.27.1
Send2Trash	1.8.0
six	1.16.0
stack-data	0.6.2
terminado	0.13.1
testpath	0.5.0
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.1.1
typing-extensions	4.3.0
urllib3	1.26.8
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.7.0

Table 7: Packages in Python 3.9.11 PBJ Workbench Runtimes - Nvidia GPU Edition

Library	Version
python	3.9.6
pip	21.1.3
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
asttokens	2.2.1

Library	Version
attrs	21.4.0
backcall	0.2.0
bitarray	2.6.0
bleach	4.1.0
certifi	2021.10.8
cff	1.15.0
charset-normalizer	2.0.11
contourpy	1.0.7
cycler	0.11.0
debugpy	1.5.1
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
executing	1.2.0
fonttools	4.39.3
idna	3.3
importlib-metadata	4.11.0
importlib-resources	5.4.0
impyla	0.18.0
ipykernel	6.9.0
ipython	8.1.0
ipython-genutils	0.2.0
jedi	0.18.1
Jinja2	3.0.3
jsonschema	4.4.0
jupyter-client	7.1.2
jupyter-kernel-gateway	2.5.1
jupyter_core	4.11.2
jupyterlab-pygments	0.1.2
kerberos	1.3.1
kiwisolver	1.4.4
MarkupSafe	2.0.1
matplotlib	3.7.0
matplotlib-inline	0.1.3
mistune	0.8.4
nbclient	0.5.10
nbconvert	6.4.2
nbformat	5.1.3
nest-asyncio	1.5.4

Library	Version
notebook	6.4.12
numpy	1.21.6
packaging	21.3
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
prometheus-client	0.13.1
prompt-toolkit	3.0.27
ptyprocess	0.7.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.21
Pygments	2.11.2
pyparsing	3.0.9
pyrsistent	0.18.1
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.27.1
Send2Trash	1.8.0
six	1.16.0
stack-data	0.6.2
terminado	0.13.1
testpath	0.5.0
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.1.1
typing_extensions	4.3.0
urllib3	1.26.8
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.7.0

Python 3.8 Libraries for Workbench

This section lists the packages in Python 3.8 Workbench Runtimes that ship with ML Runtimes 2023.05.

Table 8: Packages in Python 3.8.16 Workbench Runtimes - Standard

Library	Version
python	3.8.16
pip	23.1.2
argcomplete	2.0.0
asttokens	2.2.1
backcall	0.2.0
bitarray	2.6.0
certifi	2021.5.30
charset-normalizer	2.0.6
comm	0.1.3
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2
entrypoints	0.4
executing	1.2.0
fonttools	4.39.3
idna	3.2
importlib-metadata	4.11.3
importlib-resources	5.12.0
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
jedi	0.18.2
jupyter-client	7.2.2
jupyter_core	5.3.0
kerberos	1.3.1
kiwisolver	1.4.4
matplotlib	3.7.0
matplotlib-inline	0.1.3
nest-asyncio	1.5.5
numpy	1.21.6
packaging	23.1
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
platformdirs	3.5.0

Library	Version
prompt-toolkit	3.0.8
psutil	5.9.5
ptyprocess	0.6.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	3.0.9
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.26.0
six	1.16.0
stack-data	0.6.2
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.9.0
typing_extensions	4.3.0
urllib3	1.26.6
wcwidth	0.2.5
zipp	3.15.0

Table 9: Packages in Python 3.8.16 Workbench Runtimes - Nvidia GPU Edition

Library	Version
python	3.8.16
pip	23.1.2
cuda-libraries	11.4.1-1
libcudnn8	8.2.2.26-1
argcomplete	2.0.0
asttokens	2.2.1
backcall	0.2.0
bitarray	2.6.0
certifi	2021.5.30
charset-normalizer	2.0.6
comm	0.1.3
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2

Library	Version
entrypoints	0.4
executing	1.2.0
fonttools	4.39.3
idna	3.2
importlib-metadata	4.11.3
importlib-resources	5.12.0
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
jedi	0.18.2
jupyter-client	7.2.2
jupyter_core	5.3.0
kerberos	1.3.1
kiwisolver	1.4.4
matplotlib	3.7.0
matplotlib-inline	0.1.3
nest-asyncio	1.5.5
numpy	1.21.6
packaging	23.1
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
platformdirs	3.5.0
prompt-toolkit	3.0.8
psutil	5.9.5
ptyprocess	0.6.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	3.0.9
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.26.0
six	1.16.0
stack-data	0.6.2
thrift	0.16.0
thrift-sasl	0.4.3

Library	Version
tornado	6.1
traitlets	5.9.0
typing_extensions	4.3.0
urllib3	1.26.6
wcwidth	0.2.5
zipp	3.15.0

Table 10: Packages in Python 3.8.16 PBJ Workbench Runtimes - Standard

Library	Version
python	3.8.16
pip	23.1.2
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
asttokens	2.2.1
attrs	21.4.0
backcall	0.2.0
bitarray	2.6.0
bleach	4.1.0
certifi	2021.10.8
cffi	1.15.0
charset-normalizer	2.0.11
contourpy	1.0.7
cycler	0.11.0
debugpy	1.5.1
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
executing	1.2.0
fonttools	4.39.3
idna	3.3
importlib-metadata	4.11.0
importlib-resources	5.4.0
impyla	0.18.0
ipykernel	6.9.0
ipython	8.1.0
ipython-genutils	0.2.0
jedi	0.18.1
Jinja2	3.0.3
jsonschema	4.4.0

Library	Version
jupyter-client	7.1.2
jupyter-kernel-gateway	2.5.1
jupyter_core	4.11.2
jupyterlab-pygments	0.1.2
kerberos	1.3.1
kiwisolver	1.4.4
MarkupSafe	2.0.1
matplotlib	3.7.0
matplotlib-inline	0.1.3
mistune	0.8.4
nbclient	0.5.10
nbconvert	6.4.2
nbformat	5.1.3
nest-asyncio	1.5.4
notebook	6.4.12
numpy	1.21.6
packaging	21.3
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
prometheus-client	0.13.1
prompt-toolkit	3.0.27
ptyprocess	0.7.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.21
Pygments	2.11.2
pyparsing	3.0.9
pyrsistent	0.18.1
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.27.1
Send2Trash	1.8.0
six	1.16.0
stack-data	0.6.2
terminado	0.13.1

Library	Version
testpath	0.5.0
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.1.1
typing_extensions	4.3.0
urllib3	1.26.8
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.7.0

Table 11: Packages in Python 3.8.16 PBJ Workbench Runtimes - Nvidia GPU Edition

Library	Version
python	3.8.16
pip	23.1.2
cuda	11.8.0-1
libcudnn8	8.7.0.84-1
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
asttokens	2.2.1
attrs	21.4.0
backcall	0.2.0
bitarray	2.6.0
bleach	4.1.0
certifi	2021.10.8
cffi	1.15.0
charset-normalizer	2.0.11
contourpy	1.0.7
cycler	0.11.0
debugpy	1.5.1
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
executing	1.2.0
fonttools	4.39.3
idna	3.3
importlib-metadata	4.11.0
importlib-resources	5.4.0
impyla	0.18.0

Library	Version
ipykernel	6.9.0
ipython	8.1.0
ipython-genutils	0.2.0
jedi	0.18.1
Jinja2	3.0.3
jsonschema	4.4.0
jupyter-client	7.1.2
jupyter-kernel-gateway	2.5.1
jupyter_core	4.11.2
jupyterlab-pygments	0.1.2
kerberos	1.3.1
kiwisolver	1.4.4
MarkupSafe	2.0.1
matplotlib	3.7.0
matplotlib-inline	0.1.3
mistune	0.8.4
nbclient	0.5.10
nbconvert	6.4.2
nbformat	5.1.3
nest-asyncio	1.5.4
notebook	6.4.12
numpy	1.21.6
packaging	21.3
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
prometheus-client	0.13.1
prompt-toolkit	3.0.27
ptyprocess	0.7.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.21
Pygments	2.11.2
pyparsing	3.0.9
pyrsistent	0.18.1
python-dateutil	2.8.2

Library	Version
pyzmq	22.3.0
requests	2.27.1
Send2Trash	1.8.0
six	1.16.0
stack-data	0.6.2
terminado	0.13.1
testpath	0.5.0
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.1.1
typing_extensions	4.3.0
urllib3	1.26.8
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.7.0

Python 3.7 Libraries for Workbench

This section lists the packages in Python 3.7 Workbench Runtimes that ship with ML Runtimes 2023.05.

Table 12: Packages in Python 3.7.16 Workbench Runtimes - Standard

Library	Version
python	3.7.16
pip	23.1.2
argcomplete	2.0.0
backcall	0.2.0
bitarray	2.6.0
certifi	2021.5.30
charset-normalizer	2.0.6
cycler	0.11.0
debugpy	1.6.0
decorator	4.4.2
entrypoints	0.4
idna	3.2
importlib-metadata	4.11.3
impyla	0.18.0
ipykernel	6.5.0
ipython	7.31.1
jedi	0.17.2

Library	Version
jupyter-client	7.2.2
jupyter_core	4.12.0
kerberos	1.3.1
kiwisolver	1.4.4
matplotlib	3.3.4
matplotlib-inline	0.1.3
nest-asyncio	1.5.5
numpy	1.21.6
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	3.0.9
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.26.0
six	1.16.0
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.1.1
typing_extensions	4.3.0
urllib3	1.26.6
wcwidth	0.2.5
zipp	3.8.0

Table 13: Packages in Python 3.7.16 Workbench Runtimes - Nvidia GPU Edition

Library	Version
python	3.7.16
pip	23.1.2
cuda	11.4.1-1
libcudnn8	8.2.2.26
argcomplete	2.0.0
backcall	0.2.0

Library	Version
bitarray	2.6.0
certifi	2021.5.30
charset-normalizer	2.0.6
cycler	0.11.0
debugpy	1.6.0
decorator	4.4.2
entrypoints	0.4
idna	3.2
importlib-metadata	4.11.3
impyla	0.18.0
ipykernel	6.5.0
ipython	7.31.1
jedi	0.17.2
jupyter-client	7.2.2
jupyter_core	4.12.0
kerberos	1.3.1
kiwisolver	1.4.4
matplotlib	3.3.4
matplotlib-inline	0.1.3
nest-asyncio	1.5.5
numpy	1.21.6
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	3.0.9
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.26.0
six	1.16.0
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.1.1

Library	Version
typing_extensions	4.3.0
urllib3	1.26.6
wcwidth	0.2.5
zipp	3.8.0

Table 14: Packages in Python 3.7.16 PBJ Workbench - Standard

Library	Version
python	3.7.16
pip	23.1.2
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
attrs	21.4.0
backcall	0.2.0
bitarray	2.6.0
bleach	4.1.0
certifi	2021.10.8
cffi	1.15.0
charset-normalizer	2.0.11
cycler	0.11.0
debugpy	1.5.1
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
idna	3.3
importlib-metadata	4.11.0
importlib-resources	5.4.0
impyla	0.18.0
ipykernel	6.9.0
ipython	7.31.1
ipython-genutils	0.2.0
jedi	0.18.1
Jinja2	3.0.3
jsonschema	4.4.0
jupyter-client	7.1.2
jupyter-kernel-gateway	2.5.1
jupyter_core	4.11.2
jupyterlab-pygments	0.1.2
kerberos	1.3.1
kiwisolver	1.4.4

Library	Version
MarkupSafe	2.0.1
matplotlib	3.3.4
matplotlib-inline	0.1.3
mistune	0.8.4
nbclient	0.5.10
nbconvert	6.4.2
nbformat	5.1.3
nest-asyncio	1.5.4
notebook	6.4.12
numpy	1.21.6
packaging	21.3
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
prometheus-client	0.13.1
prompt-toolkit	3.0.27
ptyprocess	0.7.0
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.21
Pygments	2.11.2
pyparsing	3.0.9
pyrsistent	0.18.1
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.27.1
Send2Trash	1.8.0
six	1.16.0
terminado	0.13.1
testpath	0.5.0
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.1.1
typing_extensions	4.3.0
urllib3	1.26.8
wcwidth	0.2.5

Library	Version
webencodings	0.5.1
zipp	3.7.0

Table 15: Packages in Python 3.7.16 PBJ Workbench - Nvidia GPU Edition

Library	Version
python	3.7.16
pip	23.1.2
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
attrs	21.4.0
backcall	0.2.0
bitarray	2.6.0
bleach	4.1.0
certifi	2021.10.8
cffi	1.15.0
charset-normalizer	2.0.11
cycler	0.11.0
debugpy	1.5.1
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
idna	3.3
importlib-metadata	4.11.0
importlib-resources	5.4.0
impyla	0.18.0
ipykernel	6.9.0
ipython	7.31.1
ipython-genutils	0.2.0
jedi	0.18.1
Jinja2	3.0.3
jsonschema	4.4.0
jupyter-client	7.1.2
jupyter-kernel-gateway	2.5.1
jupyter_core	4.11.2
jupyterlab-pygments	0.1.2
kerberos	1.3.1
kiwisolver	1.4.4
MarkupSafe	2.0.1
matplotlib	3.3.4

Library	Version
matplotlib-inline	0.1.3
mistune	0.8.4
nbclient	0.5.10
nbconvert	6.4.2
nbformat	5.1.3
nest-asyncio	1.5.4
notebook	6.4.12
numpy	1.21.6
packaging	21.3
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
prometheus-client	0.13.1
prompt-toolkit	3.0.27
ptyprocess	0.7.0
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.21
Pygments	2.11.2
pyparsing	3.0.9
pyrsistent	0.18.1
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.27.1
Send2Trash	1.8.0
six	1.16.0
terminado	0.13.1
testpath	0.5.0
thrift	0.16.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	5.1.1
typing_extensions	4.3.0
urllib3	1.26.8
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.7.0

Scala 2.11 Libraries for Workbench

This section lists the packages in Scala 2.11.12 Workbench Runtimes that ship with ML Runtimes 2023.05.

Scala 2.11.12 Libraries for Workbench - Standard

Library	Version
org.scala-lang:scala-library	2.11.12
org.scalatest:scalatest	2.2.4
org.scala-lang:scala-reflect	2.11.12
org.scala-lang:scala-compiler	2.11.12
io.circe:circe-generic	0.11.2
io.circe:circe-parser	0.11.2
org.scalameta:scalameta	1.0.0

Python 3.10 Libraries for JupyterLab

This section lists the packages in Python 3.10 JupyterLab Runtimes that ship with ML Runtimes 2023.05.

Table 16: Packages in Python 3.10.9 JupyterLab Runtimes - Standard

Library	Version
python	3.10.9
pip	23.1.2
aiofiles	22.1.0
aiosqlite	0.19.0
anyio	3.6.2
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
arrow	1.2.3
asttokens	2.2.1
attrs	23.1.0
Babel	2.12.1
backcall	0.2.0
beautifulsoup4	4.11.2
bitarray	2.7.3
bleach	6.0.0
cattr	22.2.0
certifi	2022.12.7
cffi	1.15.1
charset-normalizer	3.1.0
colorama	0.4.6
comm	0.1.3
contourpy	1.0.7

Library	Version
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2
defusedxml	0.7.1
docstring-to-markdown	0.12
entrypoints	0.4
exceptiongroup	1.1.1
executing	1.2.0
fastjsonschema	2.16.3
fonttools	4.39.3
fqdn	1.5.1
gitdb	4.0.10
GitPython	3.1.31
idna	3.4
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
ipython-genutils	0.2.0
isoduration	20.11.0
jedi	0.18.2
jedi-language-server	0.40.0
Jinja2	3.1.2
json5	0.9.11
jsonpointer	2.3
jsonschema	4.17.3
jupyter-client	7.2.2
jupyter-events	0.6.3
jupyter-lsp	2.1.0
jupyter-server	1.23.4
jupyter-server-mathjax	0.2.6
jupyter-ydoc	0.2.4
jupyter_core	5.3.0
jupyter_server_fileid	0.9.0
jupyter_server_ydoc	0.6.1
jupyterlab	3.6.0
jupyterlab-git	0.41.0
jupyterlab-lsp	4.1.0
jupyterlab-pygments	0.2.2
jupyterlab_server	2.20.0

Library	Version
kerberos	1.3.1
kiwisolver	1.4.4
lsprotocol	2023.0.0a1
MarkupSafe	2.1.2
matplotlib	3.7.0
matplotlib-inline	0.1.3
mistune	2.0.5
nbclassic	0.5.3
nbclient	0.6.3
nbconvert	7.2.9
nbdime	3.2.0
nbformat	5.7.3
nest-asyncio	1.5.5
notebook	6.5.4
notebook_shim	0.2.2
numpy	1.24.2
packaging	23.1
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.5.0
platformdirs	3.5.0
prometheus-client	0.16.0
prompt-toolkit	3.0.8
psutil	5.9.5
ptyprocess	0.6.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.21
pydantic	1.10.7
pygls	1.0.1
Pygments	2.8.1
pyparsing	3.0.9
pyrsistent	0.19.3
python-dateutil	2.8.2
python-json-logger	2.0.7
PyYAML	6.0

Library	Version
pyzmq	22.3.0
requests	2.28.2
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1
Send2Trash	1.8.0
six	1.16.0
smmap	5.0.0
sniffio	1.3.0
soupsieve	2.4
stack-data	0.6.2
terminado	0.17.1
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.2.1
tomli	2.0.1
tornado	6.1
traitlets	5.9.0
typeguard	2.13.3
typing_extensions	4.5.0
uri-template	1.2.0
urllib3	1.26.14
wcwidth	0.2.5
webcolors	1.13
webencodings	0.5.1
websocket-client	1.5.1
y-py	0.5.9
ypy-websocket	0.8.2

Table 17: Packages in Python 3.10.9 JupyterLab Runtimes - Nvidia GPU Edition

Library	Version
python	3.10.9
pip	23.1.2
cuda	11.8.0-1
libcudnn8	8.7.0.84
aiofiles	22.1.0
aiosqlite	0.19.0
anyio	3.6.2
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0

Library	Version
arrow	1.2.3
asttokens	2.2.1
attrs	23.1.0
Babel	2.12.1
backcall	0.2.0
beautifulsoup4	4.11.2
bitarray	2.7.3
bleach	6.0.0
cattr	22.2.0
certifi	2022.12.7
ffi	1.15.1
charset-normalizer	3.1.0
colorama	0.4.6
comm	0.1.3
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2
defusedxml	0.7.1
docstring-to-markdown	0.12
entrypoints	0.4
exceptiongroup	1.1.1
executing	1.2.0
fastjsonschema	2.16.3
fonttools	4.39.3
fqdn	1.5.1
gitdb	4.0.10
GitPython	3.1.31
idna	3.4
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
ipython-genutils	0.2.0
isoduration	20.11.0
jedi	0.18.2
jedi-language-server	0.40.0
Jinja2	3.1.2
json5	0.9.11
jsonpointer	2.3

Library	Version
jsonschema	4.17.3
jupyter-client	7.2.2
jupyter-events	0.6.3
jupyter-lsp	2.1.0
jupyter-server	1.23.4
jupyter-server-mathjax	0.2.6
jupyter-ydoc	0.2.4
jupyter_core	5.3.0
jupyter_server_fileid	0.9.0
jupyter_server_ydoc	0.6.1
jupyterlab	3.6.0
jupyterlab-git	0.41.0
jupyterlab-lsp	4.1.0
jupyterlab-pygments	0.2.2
jupyterlab_server	2.20.0
kerberos	1.3.1
kiwisolver	1.4.4
lsprotocol	2023.0.0a1
MarkupSafe	2.1.2
matplotlib	3.7.0
matplotlib-inline	0.1.3
mistune	2.0.5
nbclassic	0.5.3
nbclient	0.6.3
nbconvert	7.2.9
nbdime	3.2.0
nbformat	5.7.3
nest-asyncio	1.5.5
notebook	6.5.4
notebook_shim	0.2.2
numpy	1.24.2
packaging	23.1
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.5.0
platformdirs	3.5.0
prometheus-client	0.16.0

Library	Version
prompt-toolkit	3.0.8
psutil	5.9.5
ptyprocess	0.6.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.21
pydantic	1.10.7
pygls	1.0.1
Pygments	2.8.1
pyparsing	3.0.9
pyrsistent	0.19.3
python-dateutil	2.8.2
python-json-logger	2.0.7
PyYAML	6.0
pyzmq	22.3.0
requests	2.28.2
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1
Send2Trash	1.8.0
six	1.16.0
smmap	5.0.0
sniffio	1.3.0
soupsieve	2.4
stack-data	0.6.2
terminado	0.17.1
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.2.1
tomli	2.0.1
tornado	6.1
traitlets	5.9.0
typeguard	2.13.3
typing_extensions	4.5.0
uri-template	1.2.0
urllib3	1.26.14
wcwidth	0.2.5
webcolors	1.13
webencodings	0.5.1

Library	Version
websocket-client	1.5.1
y-py	0.5.9
ypy-websocket	0.8.2

Python 3.9 Libraries for JupyterLab

This section lists the packages in Python 3.9 JupyterLab Runtimes that ship with ML Runtimes 2023.05.

Table 18: Packages in Python 3.9.16 JupyterLab Runtimes - Standard

Library	Version
python	3.9.16
pip	23.1.2
aiofiles	22.1.0
aiosqlite	0.19.0
anyio	3.3.0
argcomplete	2.0.0
argon2-cffi	20.1.0
arrow	1.2.3
asttokens	2.2.1
async-generator	1.10
attrs	23.1.0
Babel	2.12.1
backcall	0.2.0
beautifulsoup4	4.11.1
bitarray	2.6.0
bleach	3.3.1
cattrs	22.2.0
certifi	2020.11.8
cffi	1.14.4
charset-normalizer	3.1.0
colorama	0.4.6
comm	0.1.3
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2
defusedxml	0.6.0
docstring-to-markdown	0.12
entrypoints	0.4
exceptiongroup	1.1.1

Library	Version
executing	1.2.0
fastjsonschema	2.15.3
fonttools	4.39.3
fqdn	1.5.1
gitdb	4.0.10
GitPython	3.1.31
idna	2.10
importlib-metadata	4.11.3
importlib-resources	5.12.0
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
ipython-genutils	0.2.0
isoduration	20.11.0
jedi	0.18.2
jedi-language-server	0.40.0
Jinja2	3.1.1
json5	0.9.5
jsonpointer	2.3
jsonschema	4.17.3
jupyter-client	7.2.2
jupyter-events	0.6.3
jupyter-lsp	2.1.0
jupyter-server	1.24.0
jupyter-server-mathjax	0.2.6
jupyter-ydoc	0.2.4
jupyter_core	5.3.0
jupyter_server_fileid	0.9.0
jupyter_server_ydoc	0.6.1
jupyterlab	3.6.0
jupyterlab-git	0.41.0
jupyterlab-lsp	4.1.0
jupyterlab-pygments	0.1.2
jupyterlab_server	2.22.1
kerberos	1.3.1
kiwisolver	1.4.4
lsprotocol	2023.0.0a1
MarkupSafe	2.1.1
matplotlib	3.7.0

Library	Version
matplotlib-inline	0.1.3
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.5.0
nbdime	3.2.0
nbformat	5.3.0
nest-asyncio	1.5.5
notebook	6.4.12
numpy	1.21.6
packaging	23.1
pandocfilters	1.4.3
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
platformdirs	3.5.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
psutil	5.9.5
ptyprocess	0.6.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
pydantic	1.10.7
pygls	1.0.1
Pygments	2.8.1
pyparsing	3.0.9
pyrsistent	0.17.3
python-dateutil	2.8.2
python-json-logger	2.0.7
PyYAML	6.0
pyzmq	22.3.0
requests	2.29.0
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1
Send2Trash	1.8.0
six	1.16.0

Library	Version
smmap	5.0.0
sniffio	1.2.0
soupsieve	2.3.2
stack-data	0.6.2
terminado	0.9.2
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.1.1
tomli	2.0.1
tornado	6.1
traitlets	5.9.0
typeguard	2.13.3
typing_extensions	4.3.0
uri-template	1.2.0
urllib3	1.26.6
wcwidth	0.2.5
webcolors	1.13
webencodings	0.5.1
websocket-client	1.2.1
y-py	0.5.9
ypy-websocket	0.8.2
zipp	3.15.0

Table 19: Packages in Python 3.9.11 JupyterLab Runtimes - Nvidia GPU Edition

Library	Version
python	3.9.16
pip	23.1.2
cuda	11.8.0-1
libcudnn8	8.7.0.84
aiofiles	22.1.0
aiosqlite	0.19.0
anyio	3.3.0
argcomplete	2.0.0
argon2-cffi	20.1.0
arrow	1.2.3
asttokens	2.2.1
async-generator	1.10
attrs	23.1.0
Babel	2.12.1

Library	Version
backcall	0.2.0
beautifulsoup4	4.11.1
bitarray	2.6.0
bleach	3.3.1
cattr	22.2.0
certifi	2020.11.8
ffi	1.14.4
charset-normalizer	3.1.0
colorama	0.4.6
comm	0.1.3
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2
defusedxml	0.6.0
docstring-to-markdown	0.12
entrypoints	0.4
exceptiongroup	1.1.1
executing	1.2.0
fastjsonschema	2.15.3
fonttools	4.39.3
fqdn	1.5.1
gitdb	4.0.10
GitPython	3.1.31
idna	2.10
importlib-metadata	4.11.3
importlib-resources	5.12.0
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
ipython-genutils	0.2.0
isoduration	20.11.0
jedi	0.18.2
jedi-language-server	0.40.0
Jinja2	3.1.1
json5	0.9.5
jsonpointer	2.3
jsonschema	4.17.3
jupyter-client	7.2.2

Library	Version
jupyter-events	0.6.3
jupyter-lsp	2.1.0
jupyter-server	1.24.0
jupyter-server-mathjax	0.2.6
jupyter-ydoc	0.2.4
jupyter_core	5.3.0
jupyter_server_fileid	0.9.0
jupyter_server_ydoc	0.6.1
jupyterlab	3.6.0
jupyterlab-git	0.41.0
jupyterlab-lsp	4.1.0
jupyterlab-pygments	0.1.2
jupyterlab_server	2.22.1
kerberos	1.3.1
kiwisolver	1.4.4
lsprotocol	2023.0.0a1
MarkupSafe	2.1.1
matplotlib	3.7.0
matplotlib-inline	0.1.3
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.5.0
nbdime	3.2.0
nbformat	5.3.0
nest-asyncio	1.5.5
notebook	6.4.12
numpy	1.21.6
packaging	23.1
pandocfilters	1.4.3
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
platformdirs	3.5.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
psutil	5.9.5
ptyprocess	0.6.0

Library	Version
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.20
pydantic	1.10.7
pygls	1.0.1
Pygments	2.8.1
pyparsing	3.0.9
pyrsistent	0.17.3
python-dateutil	2.8.2
python-json-logger	2.0.7
PyYAML	6.0
pyzmq	22.3.0
requests	2.29.0
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1
Send2Trash	1.8.0
six	1.16.0
smmap	5.0.0
sniffio	1.2.0
soupsieve	2.3.2
stack-data	0.6.2
terminado	0.9.2
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.1.1
tomli	2.0.1
tornado	6.1
traitlets	5.9.0
typeguard	2.13.3
typing_extensions	4.3.0
uri-template	1.2.0
urllib3	1.26.6
wcwidth	0.2.5
webcolors	1.13
webencodings	0.5.1
websocket-client	1.2.1
y-py	0.5.9
ypy-websocket	0.8.2

Library	Version
zipp	3.15.0

Python 3.8 Libraries for JupyterLab

This section lists the packages in Python 3.8 JupyterLab Runtimes that ship with ML Runtimes 2023.05.

Table 20: Packages in Python 3.8.16 JupyterLab Runtimes - Standard

Library	Version
python	3.8.16
pip	23.1.2
aiofiles	22.1.0
aiosqlite	0.19.0
anyio	3.3.0
argcomplete	2.0.0
argon2-cffi	20.1.0
arrow	1.2.3
asttokens	2.2.1
async-generator	1.10
attrs	23.1.0
Babel	2.12.1
backcall	0.2.0
beautifulsoup4	4.11.1
bitarray	2.6.0
bleach	3.3.1
cattr	22.2.0
certifi	2020.11.8
cffi	1.14.4
charset-normalizer	3.1.0
colorama	0.4.6
comm	0.1.3
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2
defusedxml	0.6.0
docstring-to-markdown	0.12
entrypoints	0.4
exceptiongroup	1.1.1
executing	1.2.0
fastjsonschema	2.15.3

Library	Version
fonttools	4.39.3
fqdn	1.5.1
gitdb	4.0.10
GitPython	3.1.31
idna	2.10
importlib-metadata	4.11.3
importlib-resources	5.12.0
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
ipython-genutils	0.2.0
isoduration	20.11.0
jedi	0.18.2
jedi-language-server	0.40.0
Jinja2	3.1.1
json5	0.9.5
jsonpointer	2.3
jsonschema	4.17.3
jupyter-client	7.2.2
jupyter-events	0.6.3
jupyter-lsp	2.1.0
jupyter-server	1.24.0
jupyter-server-mathjax	0.2.6
jupyter-ydoc	0.2.4
jupyter_core	5.3.0
jupyter_server_fileid	0.9.0
jupyter_server_ydoc	0.6.1
jupyterlab	3.6.0
jupyterlab-git	0.41.0
jupyterlab-lsp	4.1.0
jupyterlab-pygments	0.1.2
jupyterlab_server	2.22.1
kerberos	1.3.1
kiwisolver	1.4.4
lsprotocol	2023.0.0a1
MarkupSafe	2.1.1
matplotlib	3.7.0
matplotlib-inline	0.1.3
mistune	0.8.4

Library	Version
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.5.0
nbdime	3.2.0
nbformat	5.3.0
nest-asyncio	1.5.5
notebook	6.4.12
numpy	1.21.6
packaging	23.1
pandocfilters	1.4.3
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
pkgutil_resolve_name	1.3.10
platformdirs	3.5.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
psutil	5.9.5
ptyprocess	0.6.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
pydantic	1.10.7
pygls	1.0.1
Pygments	2.8.1
yparsing	3.0.9
pyrsistent	0.17.3
python-dateutil	2.8.2
python-json-logger	2.0.7
pytz	2021.1
PyYAML	6.0
pyzmq	22.3.0
requests	2.29.0
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1
Send2Trash	1.8.0
six	1.16.0

Library	Version
smap	5.0.0
sniffio	1.2.0
soupsieve	2.3.2
stack-data	0.6.2
terminado	0.9.2
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.1.1
tomli	2.0.1
tornado	6.1
traitlets	5.9.0
typeguard	2.13.3
typing_extensions	4.3.0
uri-template	1.2.0
urllib3	1.26.6
wcwidth	0.2.5
webcolors	1.13
webencodings	0.5.1
websocket-client	1.2.1
y-py	0.5.9
ypy-websocket	0.8.2
zipp	3.15.0

Table 21: Packages in Python 3.8.16 JupyterLab Runtimes - Nvidia GPU Edition

Library	Version
python	3.8.16
pip	23.1.2
cuda	11.4.1-1
libcudnn8	8.7.0.84
aiofiles	22.1.0
aiosqlite	0.19.0
anyio	3.3.0
argcomplete	2.0.0
argon2-cffi	20.1.0
arrow	1.2.3
asttokens	2.2.1
async-generator	1.10
attrs	23.1.0
Babel	2.12.1

Library	Version
backcall	0.2.0
beautifulsoup4	4.11.1
bitarray	2.6.0
bleach	3.3.1
cattr	22.2.0
certifi	2020.11.8
ffi	1.14.4
charset-normalizer	3.1.0
colorama	0.4.6
comm	0.1.3
contourpy	1.0.7
cycler	0.11.0
debugpy	1.6.7
decorator	4.4.2
defusedxml	0.6.0
docstring-to-markdown	0.12
entrypoints	0.4
exceptiongroup	1.1.1
executing	1.2.0
fastjsonschema	2.15.3
fonttools	4.39.3
fqdn	1.5.1
gitdb	4.0.10
GitPython	3.1.31
idna	2.10
importlib-metadata	4.11.3
importlib-resources	5.12.0
impyla	0.18.0
ipykernel	6.22.0
ipython	8.1.0
ipython-genutils	0.2.0
isoduration	20.11.0
jedi	0.18.2
jedi-language-server	0.40.0
Jinja2	3.1.1
json5	0.9.5
jsonpointer	2.3
jsonschema	4.17.3
jupyter-client	7.2.2

Library	Version
jupyter-events	0.6.3
jupyter-lsp	2.1.0
jupyter-server	1.24.0
jupyter-server-mathjax	0.2.6
jupyter-ydoc	0.2.4
jupyter_core	5.3.0
jupyter_server_fileid	0.9.0
jupyter_server_ydoc	0.6.1
jupyterlab	3.6.0
jupyterlab-git	0.41.0
jupyterlab-lsp	4.1.0
jupyterlab-pygments	0.1.2
jupyterlab_server	2.22.1
kerberos	1.3.1
kiwisolver	1.4.4
lsprotocol	2023.0.0a1
MarkupSafe	2.1.1
matplotlib	3.7.0
matplotlib-inline	0.1.3
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.5.0
nbdime	3.2.0
nbformat	5.3.0
nest-asyncio	1.5.5
notebook	6.4.12
numpy	1.21.6
packaging	23.1
pandocfilters	1.4.3
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
pkgutil_resolve_name	1.3.10
platformdirs	3.5.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
psutil	5.9.5

Library	Version
ptyprocess	0.6.0
pure-eval	0.2.2
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
pydantic	1.10.7
pygls	1.0.1
Pygments	2.8.1
pyparsing	3.0.9
pyrsistent	0.17.3
python-dateutil	2.8.2
python-json-logger	2.0.7
pytz	2021.1
PyYAML	6.0
pyzmq	22.3.0
requests	2.29.0
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1
Send2Trash	1.8.0
six	1.16.0
smmap	5.0.0
sniffio	1.2.0
soupsieve	2.3.2
stack-data	0.6.2
terminado	0.9.2
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.1.1
tomli	2.0.1
tornado	6.1
traitlets	5.9.0
typeguard	2.13.3
typing_extensions	4.3.0
uri-template	1.2.0
urllib3	1.26.6
wcwidth	0.2.5
webcolors	1.13
webencodings	0.5.1
websocket-client	1.2.1

Library	Version
y-py	0.5.9
ypy-websocket	0.8.2
zipp	3.15.0

Python 3.7 Libraries for JupyterLab

This section lists the packages in Python 3.7 JupyterLab Runtimes that ship with ML Runtimes 2023.05.

Table 22: Packages in Python 3.7.16 JupyterLab Runtimes - Standard

Library	Version
python	3.7.16
pip	23.1.2
aiofiles	22.1.0
aiosqlite	0.19.0
anyio	3.3.0
argcomplete	2.0.0
argon2-cffi	20.1.0
arrow	1.2.3
async-generator	1.10
attrs	23.1.0
Babel	2.12.1
backcall	0.2.0
beautifulsoup4	4.11.1
bitarray	2.6.0
bleach	3.3.1
cached-property	1.5.2
certifi	2020.11.8
cffi	1.14.4
charset-normalizer	3.1.0
colorama	0.4.6
cycler	0.11.0
debugpy	1.6.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.4
fastjsonschema	2.15.3
fqdn	1.5.1
gitdb	4.0.10
GitPython	3.1.31
idna	2.10

Library	Version
importlib-metadata	4.11.3
importlib-resources	5.12.0
impyla	0.18.0
ipykernel	6.5.0
ipython	7.31.1
ipython-genutils	0.2.0
isoduration	20.11.0
jedi	0.17.2
Jinja2	3.1.1
json5	0.9.5
jsonpointer	2.3
jsonschemata	4.17.3
jupyter-client	7.2.2
jupyter-events	0.5.0
jupyter-lsp	1.5.1
jupyter-server	1.21.0
jupyter-server-mathjax	0.2.6
jupyter-ydoc	0.2.4
jupyter_core	4.12.0
jupyter_server_fileid	0.9.0
jupyter_server_ydoc	0.6.1
jupyterlab	3.6.0
jupyterlab-git	0.41.0
jupyterlab-lsp	3.10.2
jupyterlab-pygments	0.1.2
jupyterlab_server	2.22.1
kerberos	1.3.1
kiwisolver	1.4.4
MarkupSafe	2.1.1
matplotlib	3.3.4
matplotlib-inline	0.1.3
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.5.0
nbdime	3.2.0
nbformat	5.3.0
nest-asyncio	1.5.5
notebook	6.4.12

Library	Version
numpy	1.21.6
packaging	23.1
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
pkgutil_resolve_name	1.3.10
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.20
Pygments	2.8.1
pyparsing	3.0.9
pyrsistent	0.17.3
python-dateutil	2.8.2
python-json-logger	2.0.7
pytz	2021.1
PyYAML	6.0
pyzmq	22.3.0
requests	2.29.0
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1
Send2Trash	1.8.0
six	1.16.0
smap	5.0.0
sniffio	1.2.0
soupsieve	2.3.2
terminado	0.9.2
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.1.1
tomli	2.0.1
tornado	6.1
traitlets	5.1.1
typing_extensions	4.3.0
uri-template	1.2.0

Library	Version
urllib3	1.26.6
wcwidth	0.2.5
webcolors	1.13
webencodings	0.5.1
websocket-client	1.2.1
y-py	0.5.9
ypy-websocket	0.8.2
zipp	3.8.0

Table 23: Packages in Python 3.7.16 JupyterLab Runtimes - Nvidia GPU Edition

Library	Version
python	3.7.16
pip	22.1.2
cuda	11.4.1-1
aiofiles	22.1.0
aiosqlite	0.19.0
anyio	3.3.0
argcomplete	2.0.0
argon2-cffi	20.1.0
arrow	1.2.3
async-generator	1.10
attrs	23.1.0
Babel	2.12.1
backcall	0.2.0
beautifulsoup4	4.11.1
bitarray	2.6.0
bleach	3.3.1
cached-property	1.5.2
certifi	2020.11.8
cffi	1.14.4
charset-normalizer	3.1.0
colorama	0.4.6
cycler	0.11.0
debugpy	1.6.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.4
fastjsonschema	2.15.3
fqdn	1.5.1

Library	Version
gitdb	4.0.10
GitPython	3.1.31
idna	2.10
importlib-metadata	4.11.3
importlib-resources	5.12.0
impyla	0.18.0
ipykernel	6.5.0
ipython	7.31.1
ipython-genutils	0.2.0
isoduration	20.11.0
jedi	0.17.2
Jinja2	3.1.1
json5	0.9.5
jsonpointer	2.3
jsonschemata	4.17.3
jupyter-client	7.2.2
jupyter-events	0.5.0
jupyter-lsp	1.5.1
jupyter-server	1.21.0
jupyter-server-mathjax	0.2.6
jupyter-ydoc	0.2.4
jupyter_core	4.12.0
jupyter_server_fileid	0.9.0
jupyter_server_ydoc	0.6.1
jupyterlab	3.6.0
jupyterlab-git	0.41.0
jupyterlab-lsp	3.10.2
jupyterlab-pygments	0.1.2
jupyterlab_server	2.22.1
kerberos	1.3.1
kiwisolver	1.4.4
MarkupSafe	2.1.1
matplotlib	3.3.4
matplotlib-inline	0.1.3
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.5.0
nbdime	3.2.0

Library	Version
nbformat	5.3.0
nest-asyncio	1.5.5
notebook	6.4.12
numpy	1.21.6
packaging	23.1
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
pkgutil_resolve_name	1.3.10
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
Pygments	2.8.1
pyparsing	3.0.9
pyrsistent	0.17.3
python-dateutil	2.8.2
python-json-logger	2.0.7
pytz	2021.1
PyYAML	6.0
pyzmq	22.3.0
requests	2.29.0
rfc3339-validator	0.1.4
rfc3986-validator	0.1.1
Send2Trash	1.8.0
six	1.16.0
smmap	5.0.0
sniffio	1.2.0
soupsieve	2.3.2
terminado	0.9.2
thrift	0.16.0
thrift-sasl	0.4.3
tinycss2	1.1.1
tomli	2.0.1
tornado	6.1

Library	Version
traitlets	5.1.1
typing_extensions	4.3.0
uri-template	1.2.0
urllib3	1.26.6
wcwidth	0.2.5
webcolors	1.13
webencodings	0.5.1
websocket-client	1.2.1
y-py	0.5.9
ypy-websocket	0.8.2
zipp	3.8.0

R 4.0 Libraries

This section lists the R 4.0 libraries that ship with ML Runtimes 2023.05.

Table 24: PBJ R 4.0.5 Libraries for Workbench

Library	Version
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
asttokens	2.2.1
attrs	21.4.0
backcall	0.2.0
bleach	4.1.0
certifi	2021.10.8
cffi	1.15.0
charset-normalizer	2.0.11
debugpy	1.5.1
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
executing	1.2.0
idna	3.3
importlib-metadata	4.11.0
importlib-resources	5.4.0
ipykernel	6.9.0
ipython	8.1.0
ipython-genutils	0.2.0
jedi	0.18.1
Jinja2	3.0.3

Library	Version
jsonschema	4.4.0
jupyter-client	7.1.2
jupyter-core	4.11.2
jupyter-kernel-gateway	2.5.1
jupyterlab-pygments	0.1.2
MarkupSafe	2.0.1
matplotlib-inline	0.1.3
mistune	0.8.4
nbcclient	0.5.10
nbconvert	6.4.2
nbformat	5.1.3
nest-asyncio	1.5.4
notebook	6.4.12
packaging	21.3
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
prometheus-client	0.13.1
prompt-toolkit	3.0.27
ptyprocess	0.7.0
pure-eval	0.2.2
pycparser	2.21
Pygments	2.11.2
pyparsing	3.0.9
pyrsistent	0.18.1
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.27.1
Send2Trash	1.8.0
six	1.16.0
stack-data	0.6.2
terminado	0.13.1
testpath	0.5.0
tornado	6.1
traitlets	5.1.1
typing-extensions	4.3.0
urllib3	1.26.8
wcwidth	0.2.5

Library	Version
webencodings	0.5.1
zip	3.7.0

Table 25: R 4.0.5 Libraries for Workbench

Library	Version
askpass	1.1
base	4.0.5
bitops	1.0-7
boot	1.3-27
brew	1.0-8
Cairo	1.2-0
caTools	1.18.2
cdsw	2.1
class	7.3-18
cli	3.6.0
cluster	2.1.1
codetools	0.2-18
colorspace	2.1-0
compiler	4.0.5
curl	5.0.0
datasets	4.0.5
fansi	1.0.4
farver	2.1.1
foreign	0.8-81
ggplot2	3.4.0
glue	1.6.2
graphics	4.0.5
grDevices	4.0.5
grid	4.0.5
gtable	0.3.1
httr	1.4.4
isoband	0.2.7
jsonlite	1.8.4
KernSmooth	2.23-18
labeling	0.4.2
lattice	0.20-41
lifecycle	1.0.3
magrittr	2.0.3
MASS	7.3-53.1

Library	Version
Matrix	1.3-2
methods	4.0.5
mgecv	1.8-34
mime	0.12
munsell	0.5.0
nlme	3.1-152
nnet	7.3-15
openssl	2.0.5
parallel	4.0.5
pillar	1.8.1
pkgconfig	2.0.3
png	0.1-8
R6	2.5.1
RColorBrewer	1.1-3
RJSONIO	1.3-1.7
rlang	1.0.6
Rook	1.2
rpart	4.1-15
Rserve	1.7-0
scales	1.2.1
spatial	7.3-13
splines	4.0.5
stats	4.0.5
stats4	4.0.5
survival	3.2-10
sys	3.4.1
tcltk	4.0.5
tibble	3.1.8
tools	4.0.5
utf8	1.2.2
utils	4.0.5
vctrs	0.5.2
viridisLite	0.4.1
withr	2.5.0

R 4.1 Libraries

This section lists the R 4.1 libraries that ship with ML Runtimes 2023.05.

Table 26: PBJ R 4.1.1 Libraries for Workbench

Library	Version
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
asttokens	2.2.1
attrs	21.4.0
backcall	0.2.0
bleach	4.1.0
certifi	2021.10.8
cffi	1.15.0
charset-normalizer	2.0.11
debugpy	1.5.1
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
executing	1.2.0
idna	3.3
importlib-metadata	4.11.0
importlib-resources	5.4.0
ipykernel	6.9.0
ipython	8.1.0
ipython-genutils	0.2.0
jedi	0.18.1
Jinja2	3.0.3
jsonschema	4.4.0
jupyter-client	7.1.2
jupyter-core	4.11.2
jupyter-kernel-gateway	2.5.1
jupyterlab-pygments	0.1.2
MarkupSafe	2.0.1
matplotlib-inline	0.1.3
mistune	0.8.4
nbclient	0.5.10
nbconvert	6.4.2
nbformat	5.1.3
nest-asyncio	1.5.4
notebook	6.4.12
packaging	21.3
pandocfilters	1.5.0

Library	Version
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
prometheus-client	0.13.1
prompt-toolkit	3.0.27
ptyprocess	0.7.0
pure-eval	0.2.2
pycparser	2.21
Pygments	2.11.2
pyparsing	3.0.9
pyrsistent	0.18.1
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.27.1
Send2Trash	1.8.0
six	1.16.0
stack-data	0.6.2
terminado	0.13.1
testpath	0.5.0
tornado	6.1
traitlets	5.1.1
typing-extensions	4.3.0
urllib3	1.26.8
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.7.0

Table 27: R 4.1.1 Libraries for Workbench

Library	Version
askpass	1.1
base	4.1.1
bitops	1.0-7
boot	1.3-28
brew	1.0-8
Cairo	1.2-0
caTools	1.18.2
cdsw	2.1
class	7.3-19
cli	3.6.0

Library	Version
cluster	2.1.2
codetools	0.2-18
colorspace	2.1-0
compiler	4.1.1
curl	5.0.0
datasets	4.1.1
fansi	1.0.4
farver	2.1.1
foreign	0.8-81
ggplot2	3.4.0
glue	1.6.2
graphics	4.1.1
grDevices	4.1.1
grid	4.1.1
gtable	0.3.1
httr	1.4.4
isoband	0.2.7
jsonlite	1.8.4
KernSmooth	2.23-20
labeling	0.4.2
lattice	0.20-44
lifecycle	1.0.3
magrittr	2.0.3
MASS	7.3-54
Matrix	1.3-4
methods	4.1.1
mgcv	1.8-36
mime	0.12
munsell	0.5.0
nlme	3.1-152
nnet	7.3-16
openssl	2.0.5
parallel	4.1.1
pillar	1.8.1
pkgconfig	2.0.3
png	0.1-8
R6	2.5.1
RColorBrewer	1.1-3
RJSONIO	1.3-1.7

Library	Version
rlang	1.0.6
Rook	1.2
rpart	4.1-15
Rserve	1.7-0
scales	1.2.1
spatial	7.3-14
splines	4.1.1
stats	4.1.1
stats4	4.1.1
survival	3.2-11
sys	3.4.1
tcltk	4.1.1
tibble	3.1.8
tools	4.1.1
utf8	1.2.2
utils	4.1.1
vctrs	0.5.2
viridisLite	0.4.0
withr	2.5.0

R 3.6 Libraries

This section lists the R 3.6 libraries that ship with ML Runtimes 2023.05.

Table 28: PBJ R 3.6.3 Libraries for Workbench

Library	Version
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
asttokens	2.2.1
attrs	21.4.0
backcall	0.2.0
bleach	4.1.0
certifi	2021.10.8
cffi	1.15.0
charset-normalizer	2.0.11
debugpy	1.5.1
decorator	5.1.1
defusedxml	0.7.1
entrypoints	0.4
executing	1.2.0

Library	Version
idna	3.3
importlib-metadata	4.11.0
importlib-resources	5.4.0
ipykernel	6.9.0
ipython	8.1.0
ipython-genutils	0.2.0
jedi	0.18.1
Jinja2	3.0.3
jsonschema	4.4.0
jupyter-client	7.1.2
jupyter-core	4.11.2
jupyter-kernel-gateway	2.5.1
jupyterlab-pygments	0.1.2
MarkupSafe	2.0.1
matplotlib-inline	0.1.3
mistune	0.8.4
nbclient	0.5.10
nbconvert	6.4.2
nbformat	5.1.3
nest-asyncio	1.5.4
notebook	6.4.12
packaging	21.3
pandocfilters	1.5.0
parso	0.8.3
pexpect	4.8.0
pickleshare	0.7.5
prometheus-client	0.13.1
prompt-toolkit	3.0.27
ptyprocess	0.7.0
pure-eval	0.2.2
pycparser	2.21
Pygments	2.11.2
pyparsing	3.0.9
pyrsistent	0.18.1
python-dateutil	2.8.2
pyzmq	22.3.0
requests	2.27.1
Send2Trash	1.8.0
six	1.16.0

Library	Version
stack-data	0.6.2
terminado	0.13.1
testpath	0.5.0
tornado	6.1
traitlets	5.1.1
typing-extensions	4.3.0
urllib3	1.26.8
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.7.0

Table 29: R 3.6.3 Libraries for Workbench

Library	Version
askpass	1.1
base	3.6.3
bitops	1.0-7
boot	1.3-24
brew	1.0-8
Cairo	1.2-0
caTools	1.18.2
cdsw	2.1
class	7.3-15
cli	3.6.0
cluster	2.1.0
codetools	0.2-16
colorspace	2.1-0
compiler	3.6.3
curl	5.0.0
datasets	3.6.3
fansi	1.0.4
farver	2.1.1
foreign	0.8-75
ggplot2	3.4.0
glue	1.6.2
graphics	3.6.3
grDevices	3.6.3
grid	3.6.3
gtable	0.3.1
httr	1.4.4

Library	Version
isoband	0.2.7
jsonlite	1.8.4
KernSmooth	2.23-16
labeling	0.4.2
lattice	0.20-38
lifecycle	1.0.3
magrittr	2.0.3
MASS	7.3-51.5
Matrix	1.2-18
methods	3.6.3
mgcv	1.8-31
mime	0.12
munsell	0.5.0
nlme	3.1-144
nnet	7.3-12
openssl	2.0.5
parallel	3.6.3
pillar	1.8.2
pkgconfig	2.0.3
png	0.1-8
R6	2.5.1
RColorBrewer	1.1-3
RJSONIO	1.3-1.7
rlang	1.0.6
Rook	1.2
rpart	4.1-15
Rserve	1.7-0
scales	1.2.1
spatial	7.3-11
splines	3.6.3
stats	3.6.3
stats4	3.6.3
survival	3.1-8
sys	3.4.1
tcltk	3.6.3
tibble	3.1.8
tools	3.6.3
utf8	1.2.2
utils	3.6.3

Library	Version
vctr	0.5.2
viridisLite	0.4.1
withr	2.5.0

ML Runtimes 2021.09

This section lists the Python, R, and Scala libraries that ship with ML Runtimes 2021.09.



Note: Nvidia GPU Edition comes with CUDA 11.4.1 preinstalled.

Python 3.9 Libraries for Workbench

This section lists the packages in Python 3.9 Workbench Runtimes that ship with ML Runtimes 2021.09.

Table 30: Packages in Python 3.9.6 Workbench Runtimes - Standard

Library	Version
python	3.9.6
pip	21.1.3
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kerberos	1.3.1
kiwisolver	1.3.1
matplotlib	3.3.4
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1

Library	Version
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5

Table 31: Packages in Python 3.9.6 Workbench Runtimes - Nvidia GPU Edition

Library	Version
python	3.9.6
pip	21.1.3
cuda	11.4.1-1
libcudnn8	8.2.2.26-1
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kerberos	1.3.1
kiwisolver	1.3.1

Library	Version
matplotlib	3.3.4
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5

Python 3.8 Libraries for Workbench

This section lists the packages in Python 3.8 Workbench Runtimes that ship with ML Runtimes 2021.09.

Table 32: Packages in Python 3.8.6 Workbench Runtimes - Standard

Library	Version
python	3.8.6
pip	21.1.3
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4

Library	Version
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kerberos	1.3.1
kiwisolver	1.3.1
matplotlib	3.3.4
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5

Table 33: Packages in Python 3.8.6 Workbench Runtimes - Nvidia GPU Edition

Library	Version
python	3.8.6
pip	21.1.3
cuda	11.4.1-1
libcudnn8	8.2.2.26-1
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5

Library	Version
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kerberos	1.3.1
kiwisolver	1.3.1
matplotlib	3.3.4
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5

Python 3.7 Libraries for Workbench

This section lists the packages in Python 3.7 Workbench Runtimes that ship with ML Runtimes 2021.09.

Table 34: Packages in Python 3.7.11 Workbench Runtimes - Standard

Library	Version
python	3.7.11
pip	21.1.3
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kerberos	1.3.1
kiwisolver	1.3.1
matplotlib	3.3.4
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1

Library	Version
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5

Table 35: Packages in Python 3.7.11 Workbench Runtimes - Nvidia GPU Edition

Library	Version
python	3.7.11
pip	21.1.3
cuda	11.4.1-1
libcudnn8	8.2.2.26-1
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kerberos	1.3.1
kiwisolver	1.3.1
matplotlib	3.3.4
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1

Library	Version
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5

Python 3.6 Libraries for Workbench

This section lists the packages in Python 3.6 Workbench Runtimes that ship with ML Runtimes 2021.09.

Table 36: Packages in Python 3.6.14 Workbench Runtimes - Standard

Library	Version
python	3.6.14
pip	21.1.3
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kerberos	1.3.1
kiwisolver	1.3.1
matplotlib	3.3.4
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1

Library	Version
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5

Table 37: Packages in Python 3.6.14 Workbench Runtimes - Nvidia GPU Edition

Library	Version
python	3.6.14
pip	21.1.3
cuda	11.4.1-1
libcudnn8	8.2.2.26-1
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kerberos	1.3.1
kiwisolver	1.3.1

Library	Version
matplotlib	3.3.4
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5

Scala 2.11 Libraries for Workbench

This section lists the packages in Scala 2.11.12 Workbench Runtimes that ship with ML Runtimes 2021.09.

Scala 2.11.12 Libraries for Workbench - Standard

- org.scala-lang:scala-library:2.11.12
- org.scalatest:scalatest:2.2.4:test
- org.scala-lang:scala-reflect:2.11.12
- org.scala-lang:scala-compiler:2.11.12
- io.circe:circe-generic:0.11.2
- io.circe:circe-parser:0.11.2
- org.scalameta:scalameta:1.0.0

Python 3.9.6 Libraries for JupyterLab

This section lists the packages in Python 3.9 JupyterLab Runtimes that ship with ML Runtimes 2021.09.

Table 38: Packages in Python 3.9.6 JupyterLab Runtimes - Standard

Library	Version
python	3.9.6

Library	Version
pip	21.1.3
anyio	3.3.0
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.1
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.10.2
jupyterlab	3.0.17
jupyterlab-pygments	0.1.2
jupyterlab-server	2.7.2
kerberos	1.3.1
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.4
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.0.7

Library	Version
nbformat	5.1.3
nest-asyncio	1.5.1
notebook	6.4.3
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
requests-unixsocket	0.2.0
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5
webencodings	0.5.1
websocket-client	1.2.1

Table 39: Packages in Python 3.9.6 JupyterLab Runtimes - Nvidia GPU Edition

Library	Version
python	3.9.6
pip	21.1.3
cuda	11.4.1-1
libcudnn8	8.2.2.26-1
anyio	3.3.0
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.1
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.10.2
jupyterlab	3.0.17
jupyterlab-pygments	0.1.2
jupyterlab-server	2.7.2
kerberos	1.3.1
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.4

Library	Version
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.3
nest-asyncio	1.5.1
notebook	6.4.3
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
requests-unixsocket	0.2.0
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5

Library	Version
webencodings	0.5.1
websocket-client	1.2.1

Python 3.8 Libraries for JupyterLab

This section lists the packages in Python 3.8 JupyterLab Runtimes that ship with ML Runtimes 2021.09.

Table 40: Packages in Python 3.8.6 JupyterLab Runtimes - Standard

Library	Version
python	3.8.6
pip	21.1.3
anyio	3.3.0
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.1
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.10.2
jupyterlab	3.0.17
jupyterlab-pygments	0.1.2

Library	Version
jupyterlab-server	2.7.2
kerberos	1.3.1
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.4
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.3
nest-asyncio	1.5.1
notebook	6.4.3
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
requests-unixsocket	0.2.0
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0

Library	Version
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5
webencodings	0.5.1
websocket-client	1.2.1

Table 41: Packages in Python 3.8.6 JupyterLab Runtimes - Nvidia GPU Edition

Library	Version
python	3.8.6
pip	21.1.3
cuda	11.4.1-1
libcudnn8	8.2.2.26-1
anyio	3.3.0
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.1
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7

Library	Version
jupyter-core	4.6.3
jupyter-server	1.10.2
jupyterlab	3.0.17
jupyterlab-pygments	0.1.2
jupyterlab-server	2.7.2
kerberos	1.3.1
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.4
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.3
nest-asyncio	1.5.1
notebook	6.4.3
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
Pygments	2.8.1
pyarsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
requests-unixsocket	0.2.0
Send2Trash	1.5.0
six	1.15.0

Library	Version
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.6
wcwidth	0.2.5
webencodings	0.5.1
websocket-client	1.2.1

Python 3.7 Libraries for JupyterLab

This section lists the packages in Python 3.7 JupyterLab Runtimes that ship with ML Runtimes 2021.09.

Table 42: Packages in Python 3.7.11 JupyterLab Runtimes - Standard

Library	Version
python	3.7.11
pip	21.1.3
anyio	3.3.0
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.1
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
importlib-metadata	3.4.0
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1

Library	Version
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.10.2
jupyterlab	3.0.17
jupyterlab-pygments	0.1.2
jupyterlab-server	2.7.2
kerberos	1.3.1
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.4
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.3
nest-asyncio	1.5.1
notebook	6.4.3
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1

Library	Version
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
requests-unixsocket	0.2.0
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.6
wcwidth	0.2.5
webencodings	0.5.1
websocket-client	1.2.1
zipp	3.4.0

Table 43: Packages in Python 3.7.11 JupyterLab Runtimes - Nvidia GPU Edition

Library	Version
python	3.7.11
pip	21.1.3
cuda	11.4.1-1
libcudnn8	8.2.2.26-1
anyio	3.3.0
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.1
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2

Library	Version
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
importlib-metadata	3.4.0
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.10.2
jupyterlab	3.0.17
jupyterlab-pygments	0.1.2
jupyterlab-server	2.7.2
kerberos	1.3.1
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.4
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.3
nest-asyncio	1.5.1
notebook	6.4.3
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0

Library	Version
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
requests-unixsocket	0.2.0
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.6
wcwidth	0.2.5
webencodings	0.5.1
websocket-client	1.2.1
zipp	3.4.0

Python 3.6 Libraries for JupyterLab

This section lists the packages in Python 3.6 JupyterLab Runtimes that ship with ML Runtimes 2021.09.

Table 44: Packages in Python 3.6.14 JupyterLab Runtimes - Standard

Library	Version
python	3.6.14
pip	21.1.3
anyio	3.3.0
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0

Library	Version
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.1
certifi	2020.11.8
cff	1.14.4
chardet	4.0.0
contextvars	2.4
cycler	0.10.0
dataclasses	0.8
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
immutables	0.14
importlib-metadata	3.4.0
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.10.2
jupyterlab	3.0.17
jupyterlab-pygments	0.1.2
jupyterlab-server	2.7.2
kerberos	1.3.1
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.4
mistune	0.8.4
nbcassic	0.3.1
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.3
nest-asyncio	1.5.1

Library	Version
notebook	6.4.3
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
requests-unixsocket	0.2.0
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.6
wcwidth	0.2.5
webencodings	0.5.1
websocket-client	1.2.1
zipp	3.4.0

Table 45: Packages in Python 3.6.14 JupyterLab Runtimes - Nvidia GPU Edition

Library	Version
python	3.6.14
pip	21.1.3
cuda	11.4.1-1
libcudnn8	8.2.2.26-1
anyio	3.3.0
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.1
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
contextvars	2.4
cycler	0.10.0
dataclasses	0.8
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
immutables	0.14
importlib-metadata	3.4.0
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.10.2
jupyterlab	3.0.17
jupyterlab-pygments	0.1.2
jupyterlab-server	2.7.2

Library	Version
kerberos	1.3.1
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.4
mistune	0.8.4
nbclassic	0.3.1
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.3
nest-asyncio	1.5.1
notebook	6.4.3
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.3.1
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
requests-unixsocket	0.2.0
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3

Library	Version
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.6
wcwidth	0.2.5
webencodings	0.5.1
websocket-client	1.2.1
zipp	3.4.0

R 4.0 Libraries

This section lists the R 4.0 libraries that ship with ML Runtimes 2021.09.

Table 46: R 4.0.5 Libraries for Workbench

Library	Version
askpass	1.1
assertthat	0.2.1
base	4.0.5
bitops	1.0-6
boot	1.3-27
brew	1.0-6
Cairo	1.2-0
caTools	1.18.0
cdsw	2.1
class	7.3-18
cli	2.0.1
cluster	2.1.1
codetools	0.2-18
colorspace	1.4-1
compiler	4.0.5
crayon	1.3.4
curl	4.3
datasets	4.0.5
digest	0.6.23
ellipsis	0.3.0
fansi	0.4.1
farver	2.0.3
foreigns	0.8-81
ggplot2	3.2.1
glue	1.3.1

Library	Version
graphics	4.0.5
grDevices	4.0.5
grid	4.0.5
gtable	0.3.0
httr	1.4.1
jsonlite	1.6
KernSmooth	2.23-18
labeling	0.3
lattice	0.20-41
lazyeval	0.2.2
lifecycle	0.1.0
magrittr	1.5
MASS	7.3-53.1
Matrix	1.3-2
methods	4.0.5
mgecv	1.8-34
mime	0.8
munsell	0.5.0
nlme	3.1-152
nnet	7.3-15
openssl	1.4.1
parallel	4.0.5
pillar	1.4.3
pkgconfig	2.0.3
plyr	1.8.5
png	0.1-7
R6	2.4.1
RColorBrewer	1.1-2
Rcpp	1.0.3
reshape2	1.4.3
RJSONIO	1.3-1.4
rlang	0.4.4
Rook	1.1-1
rpart	4.1-15
Rserve	1.7-0
scales	1.1.0
spatial	7.3-13
splines	4.0.5
stats	4.0.5

Library	Version
stats4	4.0.5
stringi	1.4.5
stringr	1.4.0
survival	3.2-10
sys	3.3
tcltk	4.0.5
tibble	2.1.3
tools	4.0.5
utf8	1.1.4
utils	4.0.5
vctrs	0.2.2
viridisLite	0.3.0
withr	2.1.2

R 3.6 Libraries

This section lists the R 3.6 libraries that ship with ML Runtimes 2021.09.

Table 47: R 3.6.3 Libraries for Workbench

Library	Version
askpass	1.1
assertthat	0.2.1
base	3.6.3
bitops	1.0-6
boot	1.3-24
brew	1.0-6
Cairo	1.2-0
caTools	1.18.0
cdsw	2.1
class	7.3-15
cli	2.0.1
cluster	2.1.0
codetools	0.2-16
colorspace	1.4-1
compiler	3.6.3
crayon	1.3.4
curl	4.3
datasets	3.6.3
digest	0.6.23
ellipsis	0.3.0

Library	Version
fansi	0.4.1
farver	2.0.3
foreigns	0.8-75
ggplot2	3.2.1
glue	1.3.1
graphics	3.6.3
grDevices	3.6.3
grid	3.6.3
gtable	0.3.0
httr	1.4.1
jsonlite	1.6
KernSmooth	2.23-16
labeling	0.3
lattice	0.20-38
lazyeval	0.2.2
lifecycle	0.1.0
magrittr	1.5
MASS	7.3-51.5
Matrix	1.2-18
methods	3.6.3
mgcv	1.8-31
mime	0.8
munsell	0.5.0
nlme	3.4-144
nnet	7.3-12
openssl	1.4.1
parallel	3.6.3
pillar	1.4.3
pkgconfig	2.0.3
plyr	1.8.5
png	0.1-7
R6	2.4.1
RColorBrewer	1.1-2
Rcpp	1.0.3
reshape2	1.4.3
RJSONIO	1.3-1.4
rlang	0.4.4
Rook	1.1-1
rpart	4.1-15

Library	Version
Rserve	1.7-0
scales	1.1.0
spatial	7.3-11
splines	3.6.3
stats	3.6.3
stats4	3.6.3
stringi	1.4.5
stringr	1.4.0
survival	3.1-8
sys	3.3
tcltk	3.6.3
tibble	2.1.3
tools	3.6.3
utf8	1.1.4
utils	3.6.3
vctrs	0.2.2
viridisLite	0.3.0
withr	2.1.2

ML Runtimes 2021.06

This section lists the Python and R libraries that ship with ML Runtimes 2021.06.



Note: Nvidia GPU Edition comes with CUDA 11.1 preinstalled.

Python 3.8.6 Libraries for Workbench

This section lists the Python 3.8.6 libraries for Workbench that ship with ML Runtimes 2021.06.

Table 48: Python 3.8.6 Libraries for Workbench - Standard

Library	Version
backcall	0.2.0
bitarray	2.10.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4

Library	Version
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Table 49: Python 3.8.6 Libraries for Workbench - Nvidia GPU Edition

Library	Version
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0

Library	Version
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Python 3.7.9 Libraries for Workbench

This section lists the Python 3.7.9 libraries for Workbench that ship with ML Runtimes 2021.06.

Table 50: Python 3.7.9 Libraries for Workbench - Standard

Library	Version
backcall	0.2.0
bitarray	2.10.0
certifi	2020.12.5
chardet	4.0.0

Library	Version
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Table 51: Python 3.7.9 Libraries for Workbench - Nvidia GPU Edition

Library	Version
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5

Library	Version
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Python 3.6.12 Libraries for Workbench

This section lists the Python 3.6.12 Scala libraries for Workbench that ship with ML Runtimes 2021.06.

Table 52: Python 3.6.12 Libraries for Workbench - Standard

Library	Version
backcall	0.2.0
bitarray	2.10.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Table 53: Python 3.6.12 Libraries for Workbench - Nvidia GPU Edition

Library	Version
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Python 3.8.6 Libraries for JupyterLab

This section lists the Python 3.8.6 libraries for JupyterLab that ship with ML Runtimes 2021.06.

Table 54: Python 3.8.6 Libraries for JupyterLab - Standard

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bleach	3.3.0
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6
nbclient	0.5.1

Library	Version
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1

Table 55: Python 3.8.6 Libraries for JupyterLab - Nvidia GPU Edition

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.0
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.2

Library	Version
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1

Python 3.7.9 Libraries for JupyterLab

This section lists the Python 3.7.9 libraries for JupyterLab that ship with ML Runtimes 2021.06.

Table 56: Python 3.7.9 Libraries for JupyterLab - Standard

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.0
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
importlib-metadata	3.4.0
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6
nbclient	0.5.1

Library	Version
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.4.0

Table 57: Python 3.7.9 Libraries for JupyterLab - Nvidia GPU Edition

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.0
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
importlib-metadata	3.4.0
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6
nbclient	0.5.1
nbconvert	6.0.7

Library	Version
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.4.0

Python 3.6.12 Libraries for JupyterLab

This section lists the Python 3.6.12 libraries for JupyterLab that ship with ML Runtimes 2021.06.

Table 58: Python 3.6.12 Libraries for JupyterLab - Standard

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bleach	3.3.0
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
contextvars	2.4
cycler	0.10.0
dataclasses	0.8
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
immutables	0.14
importlib-metadata	3.4.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6

Library	Version
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
py4j	0.10.9.1
pycparser	2.20
Pygments	2.7.2
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.4.0

Table 59: Python 3.6.12 Libraries for JupyterLab - Nvidia GPU Edition

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bleach	3.3.0
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
contextvars	2.4
cycler	0.10.0
dataclasses	0.8
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
immutables	0.14
importlib-metadata	3.4.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jjsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6
nbclient	0.5.1

Library	Version
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
py4j	0.10.9.1
pycparser	2.20
Pygments	2.7.2
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.4.0

R 4.0 Libraries

This section lists the R 4.0 libraries that ship with ML Runtimes 2021.06.

Table 60: R 4.0 Libraries for Workbench

Library	Version
askpass	1.1
assertthat	0.2.1
base	4.0.4
bitops	1.0-6
boot	1.3-26
brew	1.0-6
Cairo	1.2-0
caTools	1.18.0
cdsw	2.1
class	7.3-18
cli	2.0.1
cluster	2.1.0
codetools	0.2-18
colorspace	1.4-1
compiler	4.0.4
crayon	1.3.4
curl	4.3
datasets	4.0.4
digest	0.6.23
ellipsis	0.3.0
fansi	0.4.1
farver	2.0.3
foreigns	0.8-81
ggplot2	3.2.1
glue	1.3.1
graphics	4.0.4
grDevices	4.0.4
grid	4.0.4
gtable	0.3.0
httr	1.4.1
jsonlite	1.6
KernSmooth	2.23-18
labeling	0.3
lattice	0.20-41
lazyeval	0.2.2
lifecycle	0.1.0
magrittr	1.5

Library	Version
MASS	7.3-53
Matrix	1.3-2
methods	4.0.4
mgcv	1.8-33
mime	0.8
munsell	0.5.0
nlme	3.1-152
nnet	7.3-15
openssl	1.4.1
parallel	4.0.4
pillar	1.4.3
pkgconfig	2.0.3
plyr	1.8.5
png	0.1-7
R6	2.4.1
RColorBrewer	1.1-2
Rcpp	1.0.3
reshape2	1.4.3
RJSONIO	1.3-1.4
rlang	0.4.4
Rook	1.1-1
rpart	4.1-15
Rserve	1.7-0
scales	1.1.0
spatial	7.3-13
splines	4.0.4
stats	4.0.4
stats4	4.0.4
stringi	1.4.5
stringr	1.4.0
survival	3.2-7
sys	3.3
tcltk	4.0.4
tibble	2.1.3
tools	4.0.4
utf8	1.1.4
utils	4.0.4
vctrs	0.2.2
viridisLite	0.3.0

Library	Version
withr	2.1.2

R 3.6 Libraries

This section lists the R 3.6 libraries that ship with ML Runtimes 2021.06.

Table 61: R 3.6 Libraries for Workbench

Library	Version
askpass	1.1
assertthat	0.2.1
base	3.6.3
bitops	1.0-6
boot	1.3-24
brew	1.0-6
Cairo	1.2-0
caTools	1.18.0
cdsw	2.1
class	7.3-15
cli	2.0.1
cluster	2.1.0
codetools	0.2-16
colorspace	1.4-1
compiler	3.6.3
crayon	1.3.4
curl	4.3
datasets	3.6.3
digest	0.6.23
ellipsis	0.3.0
fansi	0.4.1
farver	2.0.3
foreign	0.8-75
ggplot2	3.2.1
glue	1.3.1
graphics	3.6.3
grDevices	3.6.3
grid	3.6.3
gtable	0.3.0
httr	1.4.1
jsonlite	1.6
KernSmooth	2.23-16

Library	Version
labeling	0.3
lattice	0.20-38
lazyeval	0.2.2
lifecycle	0.1.0
magrittr	1.5
MASS	7.3-51.5
Matrix	1.2-18
methods	3.6.3
mgcv	1.8-31
mime	0.8
munsell	0.5.0
nlme	3.4-144
nnet	7.3-12
openssl	1.4.1
parallel	3.6.3
pillar	1.4.3
pkgconfig	2.0.3
plyr	1.8.5
png	0.1-7
R6	2.4.1
RColorBrewer	1.1-2
Rcpp	1.0.3
reshape2	1.4.3
RJSONIO	1.3-1.4
rlang	0.4.4
Rook	1.1-1
rpart	4.1-15
Rserve	1.7-0
scales	1.1.0
spatial	7.3-11
splines	3.6.3
stats	3.6.3
stats4	3.6.3
stringi	1.4.5
stringr	1.4.0
survival	3.1-8
sys	3.3
tcltk	3.6.3
tibble	2.1.3

Library	Version
tools	3.6.3
utf8	1.1.4
utils	3.6.3
vctrs	0.2.2
viridisLite	0.3.0
withr	2.1.2

ML Runtimes 2021.04

This section lists the Python and R libraries that ship with ML Runtimes 2021.04.



Note: Nvidia GPU Edition comes with CUDA 11.1 preinstalled.

RAPIDS Runtime PIP Python 3.7.8 Libraries for Workbench

This section lists the RAPIDS Runtime Python 3.8 libraries for Workbench that ship with ML Runtimes 2021.04.

Table 62: RAPIDS Runtime Python 3.7.8 Libraries for Workbench - Standard

Package	Version
aiohttp	3.7.3
appdirs	1.4.4
argon2-cffi	20.1.0
async-generator	1.10
async-timeout	3.0.1
attrs	20.3.0
bleach	3.3.0
bokeh	2.2.3
brotlipy	0.7.0
cdsw	1.0.0
certifi	2020.12.5
cffi	1.14.5
chardet	4.0.0
click	7.1.2
click-plugins	1.1.1
cligj	0.7.1
cloudpickle	1.6.0
colorcet	2.0.6
confluent-kafka	1.5.0
cryptography	3.4.4
cudf	0.18.0a0+254.g1544474166

Package	Version
cudf-kafka	0.18.0a0+254.g1544474166
cugraph	0.18.0a0+301.g0cc951f7.dirty
cuml	0.18.0a0+134.g7ca2db287
cupy	8.0.0
cusignal	0.18.0a0+33.gbc8598c
cuspatial	0.18.0a0+24.g1a2fc18
custreamz	0.18.0a0+254.g1544474166
cuxfilter	0.18.0a0+68.g236aa39
cyclcr	0.10.0
cytoolz	0.11.0
dask	2021.2.0
dask-cuda	0.18.0a210225
dask-cudf	0.18.0a0+254.g1544474166
datashader	0.11.1
datashape	0.5.4
decorator	4.4.2
defusedxml	0.6.0
distributed	2021.2.0
entrypoints	0.3
fastavro	1.3.2
fastrlock	0.5
Fiona	1.8.18
fsspec	0.8.7
GDAL	3.1.4
geopandas	0.8.1
HeapDict	1.0.1
idna	2.10
importlib-metadata	3.7.0
ipykernel	5.5.0
ipython	7.20.0
ipython-genutils	0.2.0
ipywidgets	7.6.3
jedi	0.18.0
Jinja2	2.11.3
joblib	1.0.1
jsonschema	3.2.0
jupyter-client	6.1.11
jupyter-core	4.7.1
jupyter-server-proxy	1.6.0

Package	Version
jupyterlab-pygments	0.1.2
jupyterlab-widgets	1.0.0
kiwisolver	1.3.1
llvmlite	0.35.0
loket	0.2.0
Markdown	3.3.3
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
msgpack	1.0.2
multidict	5.1.0
multipledispatch	0.6.0
munch	2.5.0
nbclient	0.5.2
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.4.3
networkx	2.5
notebook	6.2.0
numba	0.52.0
numpy	1.19.5
nvtx	0.2.3
olefile	0.46
packaging	20.9
pandas	1.1.5
pandocfilters	1.4.2
panel	0.10.3
param	1.10.1
parso	0.8.1
partd	1.1.0
pexpect	4.8.0
pickle5	0.0.11
pickleshare	0.7.5
Pillow	8.1.0
pip	21.0.1
prometheus-client	0.9.0
prompt-toolkit	3.0.16
protobuf	3.15.2
psutil	5.8.0

Package	Version
ptyprocess	0.7.0
py4j	0.10.9.1
pyarrow	1.0.1
pycparser	2.20
pyct	0.4.6
pydeck	0.5.0
pyee	7.0.4
Pygments	2.8.0
pynvml	8.0.4
pyOpenSSL	20.0.1
pyparsing	2.4.7
pypeteer	0.2.2
pyproj	2.6.1.post1
PyQt5	5.12.3
PyQt5-sip	4.19.18
PyQtChart	5.12
PyQtWebEngine	5.12.1
pyrsistent	0.17.3
PySocks	1.7.1
python-dateutil	2.8.1
pytz	2021.1
pyviz-comms	2.0.1
PyYAML	5.4.1
pyzmq	22.0.3
requests	2.25.1
rmm	0.18.0
Rtree	0.9.7
scikit-learn	0.24.1
scipy	1.6.0
Send2Trash	1.5.0
setuptools	49.6.0.post20210108
Shapely	1.7.1
simpervisor	0.4
six	1.15.0
sortedcontainers	2.3.0
streamz	0.6.2
tblib	1.6.0
terminado	0.9.2
testpath	0.4.4

Package	Version
threadpoolctl	2.1.0
toolz	0.11.1
tornado	6.1
tqdm	4.57.0
traitlets	5.0.5
treelite	1.0.0
treelite-runtime	1.0.0
typing-extensions	3.7.4.3
ucx-py	0.18.0a0+19.ge17a4dc
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1
websockets	8.1
wheel	0.36.2
widetsnbextension	3.5.1
xarray	0.16.2
xgboost	1.3.3
yaml	1.6.3
zict	2.0.0
zipp	3.4.0

RAPIDS Runtime PIP Python 3.8.6 Libraries for Workbench

This section lists the RAPIDS Runtime Python 3.8 libraries for Workbench that ship with ML Runtimes 2021.04.

Table 63: RAPIDS Runtime Python 3.7.8 Libraries for Workbench - Standard

Package	Version
aiohttp	3.7.3
appdirs	1.4.4
argon2-cffi	20.1.0
async-generator	1.10
async-timeout	3.0.1
attrs	20.3.0
backcall	0.2.0
backports.functools-lru-cache	1.6.1
bleach	3.3.0
bokeh	2.2.3
brotlipy	0.7.0
cdsw	1.0.0
certifi	2020.12.5

Package	Version
cffl	1.14.5
chardet	4.0.0
click	7.1.2
click-plugins	1.1.1
cligj	0.7.1
cloudpickle	1.6.0
colorcet	2.0.6
confluent-kafka	1.5.0
cryptography	3.4.4
cudf	0.18.0a0+254.g1544474166
cudf-kafka	0.18.0a0+254.g1544474166
cugraph	0.18.0a0+301.g0cc951f7.dirty
cuml	0.18.0a0+134.g7ca2db287
cupy	8.0.0
cusignal	0.18.0a0+33.gbc8598c
cuspatial	0.18.0a0+24.g1a2fc18
custreamz	0.18.0a0+254.g1544474166
cycler	0.10.0
cytoolz	0.11.0
dask	2021.2.0
dask-cuda	0.18.0a210225
dask-cudf	0.18.0a0+254.g1544474166
datashader	0.11.1
datashape	0.5.4
decorator	4.4.2
defusedxml	0.6.0
distributed	2021.2.0
entrypoints	0.3
fastavro	1.3.2
fastrlock	0.5
Fiona	1.8.18
fsspec	0.8.7
GDAL	3.1.4
geopandas	0.8.1
HeapDict	1.0.1
hupper	1.10.2
idna	2.10
importlib-metadata	3.7.0
ipykernel	5.5.0

Package	Version
ipython	7.20.0
ipython-genutils	0.2.0
ipywidgets	7.6.3
jedi	0.18.0
Jinja2	2.11.3
joblib	1.0.1
jsonschema	3.2.0
jupyter-client	6.1.11
jupyter-core	4.7.1
jupyter-server-proxy	1.6.0
jupyterlab-pygments	0.1.2
jupyterlab-widgets	1.0.0
kiwisolver	1.3.1
llvmlite	0.35.0
locket	0.2.0
Markdown	3.3.3
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
msgpack	1.0.2
multidict	5.1.0
multipledispatch	0.6.0
munch	2.5.0
nbclient	0.5.2
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.4.3
networkx	2.5
notebook	6.2.0
numba	0.52.0
numpy	1.19.5
nvtx	0.2.3
olefile	0.46
packaging	20.9
pandas	1.1.5
pandocfilters	1.4.2
panel	0.10.3
param	1.10.1
parso	0.8.1

Package	Version
partd	1.1.0
PasteDeploy	2.1.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
pip	21.0.1
plaster	1.0
plaster-pastedeploy	0.7
prometheus-client	0.9.0
prompt-toolkit	3.0.16
protobuf	3.15.2
psutil	5.8.0
ptyprocess	0.7.0
py4j	0.10.9.1
pyarrow	1.0.1
pyparser	2.20
pyct	0.4.6
pydeck	0.5.0
pyee	7.0.4
Pygments	2.8.0
pynvml	8.0.4
pyOpenSSL	20.0.1
pyparsing	2.4.7
pypeteer	0.2.2
pyproj	2.6.1.post1
PyQt5	5.12.3
PyQt5-sip	4.19.18
PyQtChart	5.12
PyQtWebEngine	5.12.1
pyramid	2.0
pyrsistent	0.17.3
PySocks	1.7.1
python-dateutil	2.8.1
pytz	2021.1
pyviz-comms	2.0.1
PyYAML	5.4.1
pyzmq	22.0.3
rapids	0.0.1
requests	2.25.1

Package	Version
rmm	0.18.0
Rtree	0.9.7
scikit-learn	0.24.1
scipy	1.6.0
Send2Trash	1.5.0
setuptools	49.6.0.post20210108
Shapely	1.7.1
simpervisor	0.4
six	1.15.0
sortedcontainers	2.3.0
streamz	0.6.2
tblib	1.6.0
terminado	0.9.2
testpath	0.4.4
threadpoolctl	2.1.0
toolz	0.11.1
tornado	6.1
tqdm	4.57.0
traitlets	5.0.5
translationstring	1.4
treelite	1.0.0
treelite-runtime	1.0.0
typing-extensions	3.7.4.3
ucx-py	0.18.0a0+19.ge17a4dc
urllib3	1.26.3
venusian	3.0.0
wcwidth	0.2.5
webencodings	0.5.1
WebOb	1.8.7
websockets	8.1
wheel	0.36.2
widgetsnbextension	3.5.1
xarray	0.16.2
xgboost	1.3.3
yarl	1.6.3
zict	2.0.0
zipp	3.4.0
zope.deprecation	4.4.0
zope.interface	5.4.0

RAPIDS Runtime PIP Python 3.7.8 Libraries for JupyterLab

This section lists the RAPIDS Runtime Python 3.7.8 libraries for JupyterLab that ship with ML Runtimes 2021.04.

Table 64: RAPIDS Runtime Python 3.7.8 Libraries for JupyterLab - Standard

Package	Version
aiohttp	3.7.3
anyio	2.0.2
appdirs	1.4.4
argon2-cffi	20.1.0
async-generator	1.10
async-timeout	3.0.1
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
backports.functools-lru-cache	1.6.1
bleach	3.3.0
bokeh	2.2.3
brotlipy	0.7.0
cdsw	1.0.0
certifi	2020.12.5
cffi	1.14.5
chardet	4.0.0
click	7.1.2
click-plugins	1.1.1
cligj	0.7.1
cloudpickle	1.6.0
colorcet	2.0.6
confluent-kafka	1.5.0
cryptography	3.4.4
cudf	0.18.0a0+254.g1544474166
cudf-kafka	0.18.0a0+254.g1544474166
cugraph	0.18.0a0+301.g0cc951f7.dirty
cuml	0.18.0a0+134.g7ca2db287
cupy	8.0.0
cusignal	0.18.0a0+33.gbc8598c
cuspatial	0.18.0a0+24.g1a2fc18
custreamz	0.18.0a0+254.g1544474166
cuxfilter	0.18.0a0+68.g236aa39
cycler	0.10.0
cytoolz	0.11.0

Package	Version
dask	2021.2.0
dask-cuda	0.18.0a210225
dask-cudf	0.18.0a0+254.g1544474166
datashader	0.11.1
datashape	0.5.4
decorator	4.4.2
defusedxml	0.6.0
distributed	2021.2.0
entrypoints	0.3
fastavro	1.3.2
fastrlock	0.5
Fiona	1.8.18
fsspec	0.8.7
GDAL	3.1.4
geopandas	0.8.1
HeapDict	1.0.1
idna	2.10
importlib-metadata	3.7.0
ipykernel	5.5.0
ipython	7.20.0
ipython-genutils	0.2.0
ipywidgets	7.6.3
jedi	0.18.0
Jinja2	2.11.3
joblib	1.0.1
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.11
jupyter-core	4.7.1
jupyter-server	1.2.2
jupyter-server-proxy	1.6.0
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
jupyterlab-widgets	1.0.0
kiwisolver	1.3.1
llvmlite	0.35.0
locket	0.2.0
Markdown	3.3.3

Package	Version
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
msgpack	1.0.2
multidict	5.1.0
multipledispatch	0.6.0
munch	2.5.0
nbclassic	0.2.6
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.4.3
networkx	2.5
notebook	6.2.0
numba	0.52.0
numpy	1.19.5
nvtx	0.2.3
olefile	0.46
packaging	20.9
pandas	1.1.5
pandocfilters	1.4.2
panel	0.10.3
param	1.10.1
parso	0.8.1
partd	1.1.0
pexpect	4.8.0
pickle5	0.0.11
pickleshare	0.7.5
Pillow	8.1.0
pip	21.0.1
prometheus-client	0.9.0
prompt-toolkit	3.0.16
protobuf	3.15.2
psutil	5.8.0
ptyprocess	0.7.0
py4j	0.10.9.1
pyarrow	1.0.1
pyparser	2.20
pyct	0.4.6

Package	Version
pydeck	0.5.0
pyee	7.0.4
Pygments	2.8.0
pynvml	8.0.4
pyOpenSSL	20.0.1
pyparsing	2.4.7
pypeteer	0.2.2
pyproj	2.6.1.post1
PyQt5	5.12.3
PyQt5-sip	4.19.18
PyQtChart	5.12
PyQtWebEngine	5.12.1
pyrsistent	0.17.3
PySocks	1.7.1
python-dateutil	2.8.1
pytz	2021.1
pyviz-comms	2.0.1
PyYAML	5.4.1
pymzmq	22.0.3
requests	2.25.1
rmm	0.18.0
Rtree	0.9.7
scikit-learn	0.24.1
scipy	1.6.0
Send2Trash	1.5.0
setuptools	49.6.0.post20210108
Shapely	1.7.1
simpervisor	0.4
six	1.15.0
sniffio	1.2.0
sortedcontainers	2.3.0
streamz	0.6.2
tblib	1.6.0
terminado	0.9.2
testpath	0.4.4
threadpoolctl	2.1.0
toolz	0.11.1
tornado	6.1
tqdm	4.57.0

Package	Version
traitlets	5.0.5
treelite	1.0.0
treelite-runtime	1.0.0
typing-extensions	3.7.4.3
ucx-py	0.18.0a0+19.ge17a4dc
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1
websockets	8.1
wheel	0.36.2
widetsnbextension	3.5.1
xarray	0.16.2
xgboost	1.3.3
yaml	1.6.3
zict	2.0.0
zipp	3.4.0

RAPIDS Runtime PIP Python 3.8.6 Libraries for JupyterLab

This section lists the RAPIDS Runtime Python 3.8.6 libraries for JupyterLab that ship with ML Runtimes 2021.04.

Table 65: RAPIDS Runtime Python 3.86 Libraries for JupyterLab - Standard

Package	Version
aiohttp	3.7.3
anyio	2.0.2
appdirs	1.4.4
argon2-cffi	20.1.0
async-generator	1.10
async-timeout	3.0.1
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
backports.functools-lru-cache	1.6.1
bleach	3.3.0
bokeh	2.2.3
brotlipy	0.7.0
cdsw	1.0.0
certifi	2020.12.5
cffi	1.14.5
chardet	4.0.0

Package	Version
click	7.1.2
click-plugins	1.1.1
cligj	0.7.1
cloudpickle	1.6.0
colorcet	2.0.6
confluent-kafka	1.5.0
cryptography	3.4.4
cudf	0.18.0a0+254.g1544474166
cudf-kafka	0.18.0a0+254.g1544474166
cugraph	0.18.0a0+301.g0cc951f7.dirty
cuml	0.18.0a0+134.g7ca2db287
cupy	8.0.0
cusignal	0.18.0a0+33.gbc8598c
cuspatial	0.18.0a0+24.g1a2fc18
custreamz	0.18.0a0+254.g1544474166
cycler	0.10.0
cytoolz	0.11.0
dask	2021.2.0
dask-cuda	0.18.0a210225
dask-cudf	0.18.0a0+254.g1544474166
datashader	0.11.1
datashape	0.5.4
decorator	4.4.2
defusedxml	0.6.0
distributed	2021.2.0
entrypoints	0.3
fastavro	1.3.2
fastrlock	0.5
Fiona	1.8.18
fsspec	0.8.7
GDAL	3.1.4
geopandas	0.8.1
HeapDict	1.0.1
hupper	1.10.2
idna	2.10
importlib-metadata	3.7.0
ipykernel	5.5.0
ipython	7.20.0
ipython-genutils	0.2.0

Package	Version
ipywidgets	7.6.3
jedi	0.18.0
Jinja2	2.11.3
joblib	1.0.1
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.11
jupyter-core	4.7.1
jupyter-server	1.2.2
jupyter-server-proxy	1.6.0
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
jupyterlab-widgets	1.0.0
kiwisolver	1.3.1
llvmlite	0.35.0
locket	0.2.0
Markdown	3.3.3
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
msgpack	1.0.2
multidict	5.1.0
multipledispatch	0.6.0
munch	2.5.0
nbclassic	0.2.6
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.4.3
networkx	2.5
notebook	6.2.0
numba	0.52.0
numpy	1.19.5
nvtx	0.2.3
olefile	0.46
packaging	20.9
pandas	1.1.5
pandocfilters	1.4.2

Package	Version
panel	0.10.3
param	1.10.1
parso	0.8.1
partd	1.1.0
PasteDeploy	2.1.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
pip	21.0.1
plaster	1.0
plaster-pastedeploy	0.7
prometheus-client	0.9.0
prompt-toolkit	3.0.16
protobuf	3.15.2
psutil	5.8.0
ptyprocess	0.7.0
py4j	0.10.9.1
pyarrow	1.0.1
pyparser	2.20
pyct	0.4.6
pydeck	0.5.0
pyee	7.0.4
Pygments	2.8.0
pynvml	8.0.4
pyOpenSSL	20.0.1
pyparsing	2.4.7
pypeteer	0.2.2
pyproj	2.6.1.post1
PyQt5	5.12.3
PyQt5-sip	4.19.18
PyQtChart	5.12
PyQtWebEngine	5.12.1
pyramid	2.0
pyrsistent	0.17.3
PySocks	1.7.1
python-dateutil	2.8.1
pytz	2021.1
pyviz-comms	2.0.1
PyYAML	5.4.1

Package	Version
pyzmq	22.0.3
rapids	0.0.1
requests	2.25.1
rmm	0.18.0
Rtree	0.9.7
scikit-learn	0.24.1
scipy	1.6.0
Send2Trash	1.5.0
setuptools	49.6.0.post20210108
Shapely	1.7.1
simpervisor	0.4
six	1.15.0
sniffio	1.2.0
sortedcontainers	2.3.0
streamz	0.6.2
tblib	1.6.0
terminado	0.9.2
testpath	0.4.4
threadpoolctl	2.1.0
toolz	0.11.1
tornado	6.1
tqdm	4.57.0
traitlets	5.0.5
translationstring	1.4
treelite	1.0.0
treelite-runtime	1.0.0
typing-extensions	3.7.4.3
ucx-py	0.18.0a0+19.ge17a4dc
urllib3	1.26.3
venusian	3.0.0
wcwidth	0.2.5
webencodings	0.5.1
WebOb	1.8.7
websockets	8.1
wheel	0.36.2
widetsnbextension	3.5.1
xarray	0.16.2
xgboost	1.3.3
yarl	1.6.3

Package	Version
zict	2.0.0
zipp	3.4.0
zope.deprecation	4.4.0
zope.interface	5.4.0

ML Runtimes 2021.02

This section lists the Python and R libraries that ship with ML Runtimes 2021.02.



Note: Nvidia GPU Edition comes with CUDA 11.1 preinstalled.

Python 3.8 Libraries for Workbench

This section lists the Python 3.8 libraries for Workbench that ship with ML Runtimes 2021.02.

Table 66: Python 3.8.6 Libraries for Workbench - Standard

Library	Version
backcall	0.2.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
py4j	0.10.9.1
Pygments	2.7.2

Library	Version
pparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Table 67: Python 3.8.6 Libraries for Workbench - Nvidia GPU Edition

Library	Version
backcall	0.2.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
py4j	0.10.9.1
Pygments	2.7.2
pparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1

Library	Version
six	1.15.0
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Python 3.7 Libraries for Workbench

This section lists the Python 3.7 libraries for Workbench that ship with ML Runtimes 2021.02.

Table 68: Python 3.7.9 Libraries for Workbench - Standard

Library	Version
backcall	0.2.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
py4j	0.10.9.1
Pygments	2.7.2
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0

Library	Version
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Table 69: Python 3.7.9 Libraries for Workbench - Nvidia GPU Edition

Library	Version
backcall	0.2.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
py4j	0.10.9.1
Pygments	2.7.2
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Python 3.6 Libraries for Workbench

This section lists the Python 3.6 Scala libraries for Workbench that ship with ML Runtimes 2021.02.

Table 70: Python 3.6.12 Libraries for Workbench - Standard

Library	Version
backcall	0.2.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
py4j	0.10.9.1
Pygments	2.7.2
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Table 71: Python 3.6.12 Libraries for Workbench - Nvidia GPU Edition

Library	Version
backcall	0.2.0
bitarray	2.1.0
certifi	2020.12.5
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
jupyter-client	6.1.7
jupyter-core	4.6.3
kiwisolver	1.3.1
matplotlib	3.3.2
numpy	1.19.4
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
Pygments	2.8.1
pyparsing	2.4.7
python-dateutil	2.8.1
pyzmq	19.0.2
requests	2.25.1
six	1.15.0
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5

Python 3.8 Libraries for JupyterLab

This section lists the Python 3.8 libraries for JupyterLab that ship with ML Runtimes 2021.02.

Table 72: Python 3.8.6 Libraries for JupyterLab - Standard

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bleach	3.3.0
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6
nbclient	0.5.1

Library	Version
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.2.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pycparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1

Table 73: Python 3.8.6 Libraries for JupyterLab - Nvidia GPU Edition

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bleach	3.3.0
certifi	2020.11.8
cffi	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0

Library	Version
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
py4j	0.10.9.1
pycparser	2.20
Pygments	2.7.2
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
tornado	6.1
traitlets	4.3.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1

Python 3.7 Libraries for JupyterLab

This section lists the Python 3.7 libraries for JupyterLab that ship with ML Runtimes 2021.02.

Table 74: Python 3.7.9 Libraries for JupyterLab - Standard

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0

Library	Version
Babel	2.9.0
backcall	0.2.0
bleach	3.3.0
certifi	2020.11.8
cff	1.14.4
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
importlib-metadata	3.4.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1

Library	Version
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
py4j	0.10.9.1
pycparser	2.20
Pygments	2.7.2
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.4.0

Table 75: Python 3.7.9 Libraries for JupyterLab - Nvidia GPU Edition

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bleach	3.3.0
certifi	2020.11.8
cffi	1.14.4

Library	Version
chardet	4.0.0
cycler	0.10.0
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
importlib-metadata	3.4.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbcClassic	0.2.6
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8

Library	Version
ptyprocess	0.6.0
py4j	0.10.9.1
pyparser	2.20
Pygments	2.7.2
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.4.0

Python 3.6 Libraries for JupyterLab

This section lists the Python 3.6 libraries for JupyterLab that ship with ML Runtimes 2021.02.

Table 76: Python 3.6.12 Libraries for JupyterLab - Standard

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bitarray	2.1.0
bleach	3.3.0
certifi	2020.12.8
cffi	1.14.4
chardet	4.0.0

Library	Version
contextvars	2.4
cycler	0.10.0
dataclasses	0.8
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
immutables	0.14
importlib-metadata	3.4.0
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1
pexpect	4.8.0
pickleshare	0.7.5

Library	Version
Pillow	8.2.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
pure-sasl	0.6.2
py4j	0.10.9.1
pyparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.4.0

Table 77: Python 3.6.12 Libraries for JupyterLab - Nvidia GPU Edition

Library	Version
anyio	2.0.2
argon2-cffi	20.1.0
async-generator	1.10
attrs	20.3.0
Babel	2.9.0
backcall	0.2.0
bleach	3.3.0
certifi	2020.11.8

Library	Version
cff	1.14.4
chardet	4.0.0
contextvars	2.4
cycler	0.10.0
dataclasses	0.8
decorator	4.4.2
defusedxml	0.6.0
entrypoints	0.3
idna	2.10
immutable	0.14
importlib-metadata	3.4.0
impyla	0.17.0
ipykernel	5.3.4
ipython	7.16.1
ipython-genutils	0.2.0
jedi	0.17.2
Jinja2	2.11.3
json5	0.9.5
jsonschema	3.2.0
jupyter-client	6.1.7
jupyter-core	4.6.3
jupyter-server	1.2.2
jupyterlab	3.0.6
jupyterlab-pygments	0.1.2
jupyterlab-server	2.1.3
kiwisolver	1.3.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mistune	0.8.4
nbclassic	0.2.6
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.1.2
nest-asyncio	1.5.1
notebook	6.2.0
numpy	1.19.4
packaging	20.9
pandocfilters	1.4.3
parso	0.7.1

Library	Version
pexpect	4.8.0
pickleshare	0.7.5
Pillow	8.1.0
prometheus-client	0.9.0
prompt-toolkit	3.0.8
ptyprocess	0.6.0
py4j	0.10.9.1
pyparser	2.20
Pygments	2.8.1
pyparsing	2.4.7
pyrsistent	0.17.3
python-dateutil	2.8.1
pytz	2021.1
pyzmq	19.0.2
requests	2.25.1
Send2Trash	1.5.0
six	1.15.0
sniffio	1.2.0
terminado	0.9.2
testpath	0.4.4
thrift	0.11.0
thrift-sasl	0.4.3
tornado	6.1
traitlets	4.3.3
typing-extensions	3.7.4.3
urllib3	1.26.3
wcwidth	0.2.5
webencodings	0.5.1
zipp	3.4.0

R 4.0 Libraries

This section lists the R 4.0 libraries that ship with ML Runtimes 2021.02.

Table 78: R 4.0 Libraries for Workbench

Library	Version
askpass	1.1
assertthat	0.2.1
base	4.0.3
bitops	1.0-6

Library	Version
boot	1.3-25
brew	1.0-6
Cairo	1.2-0
caTools	1.18.0
cds	2.1
class	7.3-17
cli	2.0.1
cluster	2.1.0
codetools	0.2-16
colorspace	1.4-1
compiler	4.0.3
crayon	1.3.4
curl	4.3
datasets	4.0.3
digest	0.6.23
ellipsis	0.3.0
fansi	0.4.1
farver	2.0.3
foreign	0.8-80
ggplot2	3.2.1
glue	1.3.1
graphics	4.0.3
grDevices	4.0.3
grid	4.0.3
gtable	0.3.0
httr	1.4.1
jsonlite	1.6
KernSmooth	2.23-17
labeling	0.3
lattice	0.20-41
lazyeval	0.2.2
lifecycle	0.1.0
magrittr	1.5
MASS	7.3-53
Matrix	1.2-18
methods	4.0.3
mgcv	1.8-33
mime	0.8
munsell	0.5.0

Library	Version
nlme	3.1-149
nnet	7.3-14
openssl	1.4.1
parallel	4.0.3
pillar	1.4.3
pkgconfig	2.0.3
plyr	1.8.5
png	0.1-7
R6	2.4.1
RColorBrewer	1.1-2
Rcpp	1.0.3
reshape2	1.4.3
RJSONIO	1.3-1.4
rlang	0.4.4
Rook	1.1-1
rpart	4.1-15
Rserve	1.7-0
scales	1.1.0
spatial	7.3-12
splines	4.0.3
stats	4.0.3
stats4	4.0.3
stringi	1.4.5
stringr	1.4.0
survival	3.2-7
sys	3.3
tcltk	4.0.3
tibble	2.1.3
tools	4.0.3
utf8	1.1.4
utils	4.0.3
vctrs	0.2.2
viridisLite	0.3.0
withr	2.1.2

R 3.6 Libraries

This section lists the R 3.6 libraries that ship with ML Runtimes 2021.02.

Table 79: R 3.6 Libraries for Workbench

Library	Version
askpass	1.1
assertthat	0.2.1
base	3.6.3
bitops	1.0-6
boot	1.3-24
brew	1.0-6
Cairo	1.2-0
caTools	1.18.0
cdsw	2.1
class	7.3-15
cli	2.0.1
cluster	2.1.0
codetools	0.2-16
colorspace	1.4-1
compiler	3.6.3
crayon	1.3.4
curl	4.3
datasets	3.6.3
digest	0.6.23
ellipsis	0.3.0
fansi	0.4.1
farver	2.0.3
foreigns	0.8-75
ggplot2	3.2.1
glue	1.3.1
graphics	3.6.3
grDevices	3.6.3
grid	3.6.3
gtable	0.3.0
httr	1.4.1
jsonlite	1.6
KernSmooth	2.23-16
labeling	0.3
lattice	0.20-38
lazyeval	0.2.2
lifecycle	0.1.0
magrittr	1.5

Library	Version
MASS	7.3-51.5
Matrix	1.2-18
methods	3.6.3
mgcv	1.8-31
mime	0.8
munsell	0.5.0
nlme	3.4-144
nnet	7.3-12
openssl	1.4.1
parallel	3.6.3
pillar	1.4.3
pkgconfig	2.0.3
plyr	1.8.5
png	0.1-7
R6	2.4.1
RColorBrewer	1.1-2
Rcpp	1.0.3
reshape2	1.4.3
RJSONIO	1.3-1.4
rlang	0.4.4
Rook	1.1-1
rpart	4.1-15
Rserve	1.7-0
scales	1.1.0
spatial	7.3-11
splines	3.6.3
stats	3.6.3
stats4	3.6.3
stringi	1.4.5
stringr	1.4.0
survival	3.1-8
sys	3.3
tcltk	3.6.3
tibble	2.1.3
tools	3.6.3
utf8	1.1.4
utils	3.6.3
vctrs	0.2.2
viridisLite	0.3.0

Library	Version
withr	2.1.2

ML Runtimes 2020.11

This section lists the Python and R libraries that ship with ML Runtimes 2020.11.

All ML Runtimes 2020.11 contain NumPy 1.19.2 and Matplotlib 3.1.2.

- Matplotlib - 3.1.2
- NumPy - 1.19.2 and the Python versions, which are 3.6.10, 3.7.9, and 3.8.3