Published in Analytics Vidhya

B. Chen Follow

Mar 3, 2020 · 3 min read · Member-only · Listen

Create Virtual Environment using “conda” and add it to Jupyter Notebook

Are you using anaconda and working with Jupyter Notebook and Python? In this article you will see how to create virtual environment using conda and add it to Jupyter Notebook.

Anaconda is a Python (and R) distribution that has the goal to simplify package management and deployment for scientific computing. **Jupyter Notebook** is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Here is the article outline:

- Part 1: Create Virtual Environment using “conda”
- Part 2: Add Virtual Environment to Jupyter Notebook
- Part 3: Remove Virtual Environment from Jupyter Notebook

Part 1: Create Virtual Environment using “conda”

Create a Virtual Environment

Let’s have a look how to create a virtual environment with Anaconda. After the installation you can create the conda virtual environment with:

Here is an example to create a Virtual Environment “**d2l**”

```
⇒  conda create -n d2l -y
Collecting package metadata (repodata.json): done
Solving environment: done


## Package Plan ##


environment location: /Users/chen5/anaconda3/envs/d2l


Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate d2l
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

Create an environment with a specific version of Python

```
conda create -n <my_env_name> python=3.6
```

```
conda activate <my_env_name>
```

Create an environment from an environment.yml file

Here is an example of environment.yml

```
name: d2l
dependencies:
- python=3.6
- pip:
  - mxnet==1.5.0
  - d2lzh==0.8.11
  - jupyter==1.0.0
  - matplotlib==2.2.2
  - pandas==0.23.4
```

The first line of the `yml` file sets the new environment's name. To create an environment from a **environment.yml**, you can run

```
conda env create -f environment.yml
```




You can verify quickly you are in the environment by running `which python` or `which pip` which will return the path of the python executable in the environment if all went well

```
⇒  which python
/Users/admin/anaconda3/envs/d2l/bin/python

⇒  which pip
/Users/admin/anaconda3/envs/d2l/bin/pip
```

List all available virtual environment

To list all available virtual environment and the one currently in active is marked with `*`

```
⇒  conda env list
# conda environments:
#
base                     /Users/admin/anaconda3
d2l                   *  /Users/admin/anaconda3/envs/d2l
tf                       /Users/admin/anaconda3/envs/tf
```

```
conda deactivate
```

Remove an environment

To remove an environment you can type:

```
conda env remove -n <my_env_name>
```

Part 2: Add Virtual Environment to Jupyter Notebook

Jupyter Notebook makes sure that the IPython kernel is available, but you have to manually add a kernel with a different version of Python or a virtual environment.

First, you need to activate your virtual environment. Next, install ipykernel which provides the IPython kernel for Jupyter:

```
pip install --user ipykernel
```

```
python -m ipykernel install --user --name=<my_env_name>
```

For example: add virtual environment “**d2l**” to Jupyter and it should print the following:


```
⇒  python -m ipykernel install --user --name=d2l
Installed kernelspec d2l in /Users/admin/Library/Jupyter/kernels/d2l
```

After that, you could `cd` into the env folder and inspect the configuration file

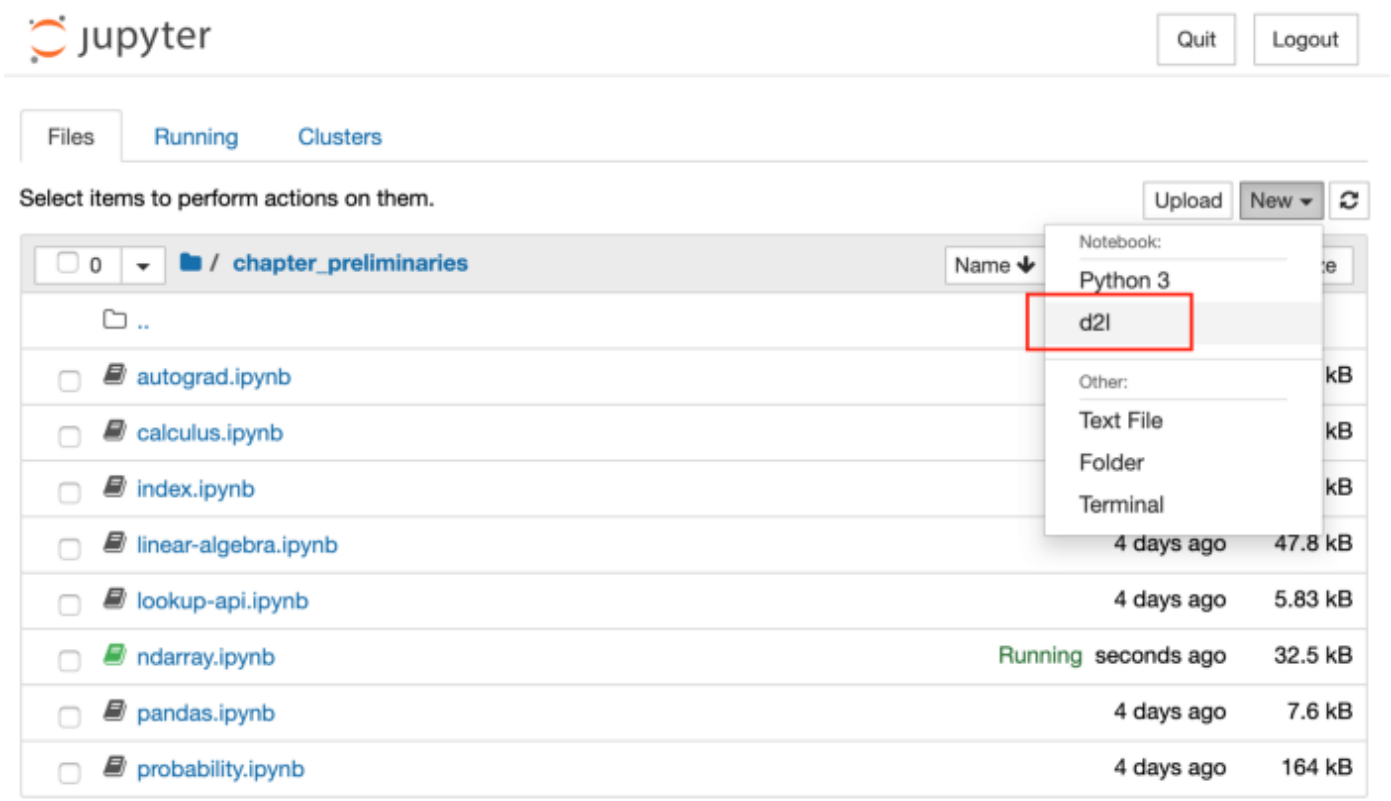
`kernel.json`

```
⇒  cd /Users/admin/Library/Jupyter/kernels/d2l
admin@MAC-156893:~/Library/Jupyter/kernels/d2l
⇒  ls
kernel.json    logo-32x32.png logo-64x64.png
admin@MAC-156893:~/Library/Jupyter/kernels/d2l
⇒  cat kernel.json
{
 "argv": [
  "/Users/admin/anaconda3/envs/d2l/bin/python",
  "-m",
  "ipykernel_launcher",
```

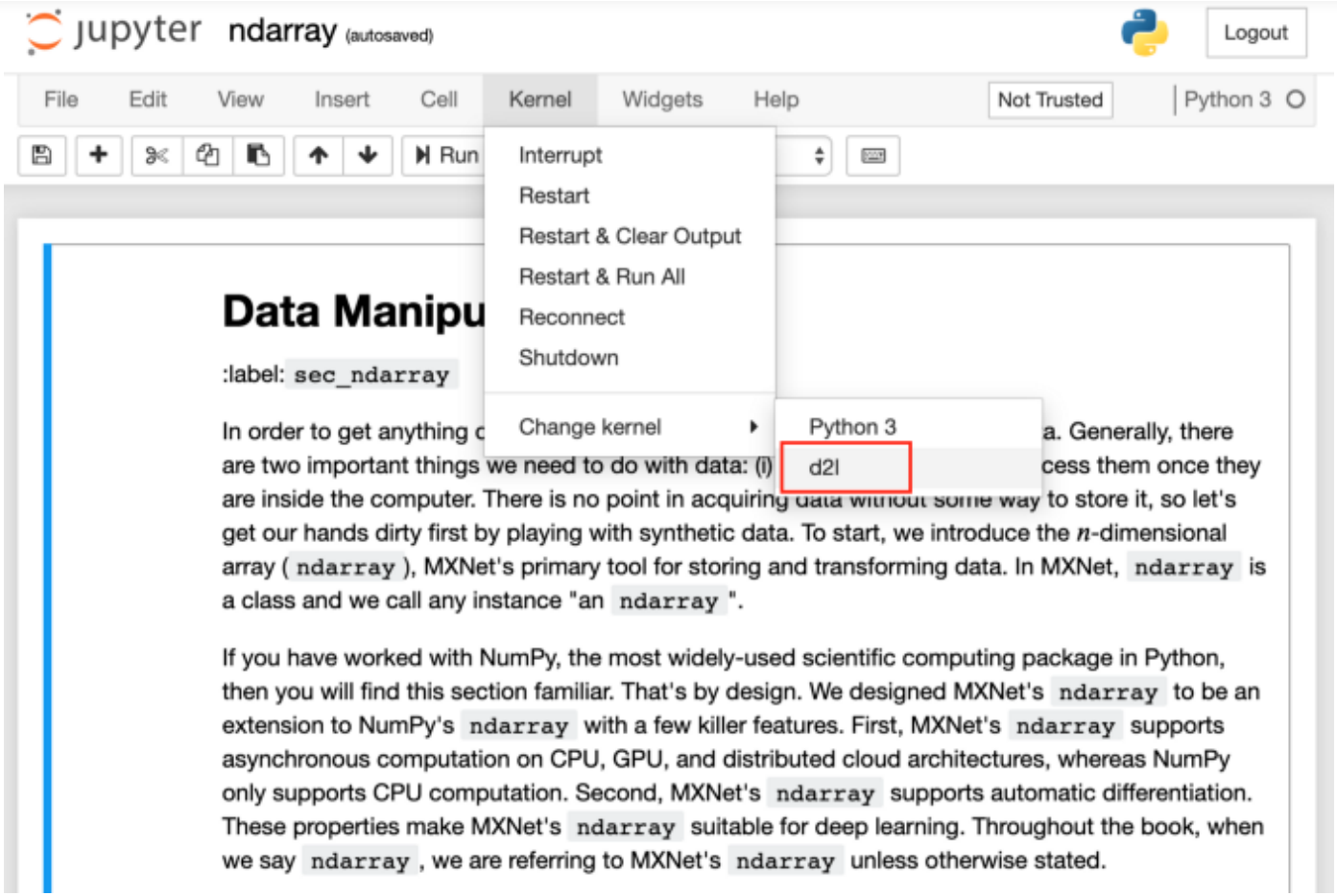
```
17    }%
```

cat-kernel.sh hosted with ❤ by GitHub view raw

Now you are able to choose the conda environment as a kernel in Jupyter Notebook. Here is what that would look like:



Select a Virtual Environment from Jupyter Notebook



Change Virtual Environment in an opened notebook

Part 3: Remove Virtual Environment from Jupyter Notebook

After you deleted your virtual environment, you'll want to remove it also from Jupyter. Let's first see which kernels are available. You can list them with:

```
  python3
/Users/admin/anaconda3/envs/d2l/share/jupyter/kernels/python3
```

Now, to uninstall the kernel, you can type:

```
jupyter kernelspec uninstall <my_env_name>
```

Enjoy!

And that's about it. Thanks for reading.