Create your workspaces

PRACTICAL

Follow the **next slides** to create your workspaces and to copy in there data and scripts necessary for the exercises.

Windows: Please be careful, since copy-pasting text from pdf might be inaccurate.

Linux: **Ctrl+C** from **PDF** and **right-click** to paste in **terminal** should work.

Create your workspaces

Execute only once!

Once you have logged in, create a workspace for each course day (with **name** *wsdayX* and **duration** *15 days* **or until the end of the course**):

```
> ws_allocate  wsday1  5
> ws_allocate  wsday2  5
> ws_allocate  wsday3  5
```

Name might be different in the screenshots

Workspace: (Very) high **disk storage** (quota)!
Home-directory: Limited quota!

To navigate to a workspace:

Recover the workspace id with ws_list:

```
> ws_list
```

Define the **path** variable MYSCR:

```
> MYSCR=$(ws_find  wsday1)
```

Navigate to your workspace:

```
> cd  $MYSCR
```

```
s28349 cl5fr2 28$ ws_list
```

```
id: wsday1
     workspace directory  : /lustre/nec/ws2/ws
     remaining time       : 14 days 23 hours
     creation time        : Wed Jun 23 11:37:1
     expiration date      : Thu Jul  8 11:37:1
     filesystem name      : NEC_lustre
     available extensions : 3
```

```
s28349 cl5fr3 33$ MYSCR=$(ws_find wsday1)
```

https://kb.hlrs.de/platforms/index.php/Workspace_mechanism

Create your workspaces (online course)

MAC-Users:

You might encounter the error:

> MYSCR=$(ws_allocate *wsMyWorkspace* 15)

Info: creating workspace.

Error: could not create workspace directory!

In that case a solution is e.g.

https://www.cyberciti.biz/faq/os-x-terminal-bash-warning-setlocale-lc_ctype-cannot-change-locale/

(LC_CTYPE should not be UTF-8 but e.g. en_US.UTF-8)

Copy and extract all data for day 1

Navigate to your day 1 workspace:

```
> MYSCR=$(ws_find  wsday1)
> cd  $MYSCR
```

… and copy the following archives (do not forget the point at the end):

```
> cp /shared/akad-dl-hlrs/day1/dl-hlrs1.tar.gz  .
```

Extract the content of the *tar* file:

```
> tar  -xzf  dl-hlrs1.tar.gz
```

Check that all folders are there with:

```
> ls
```

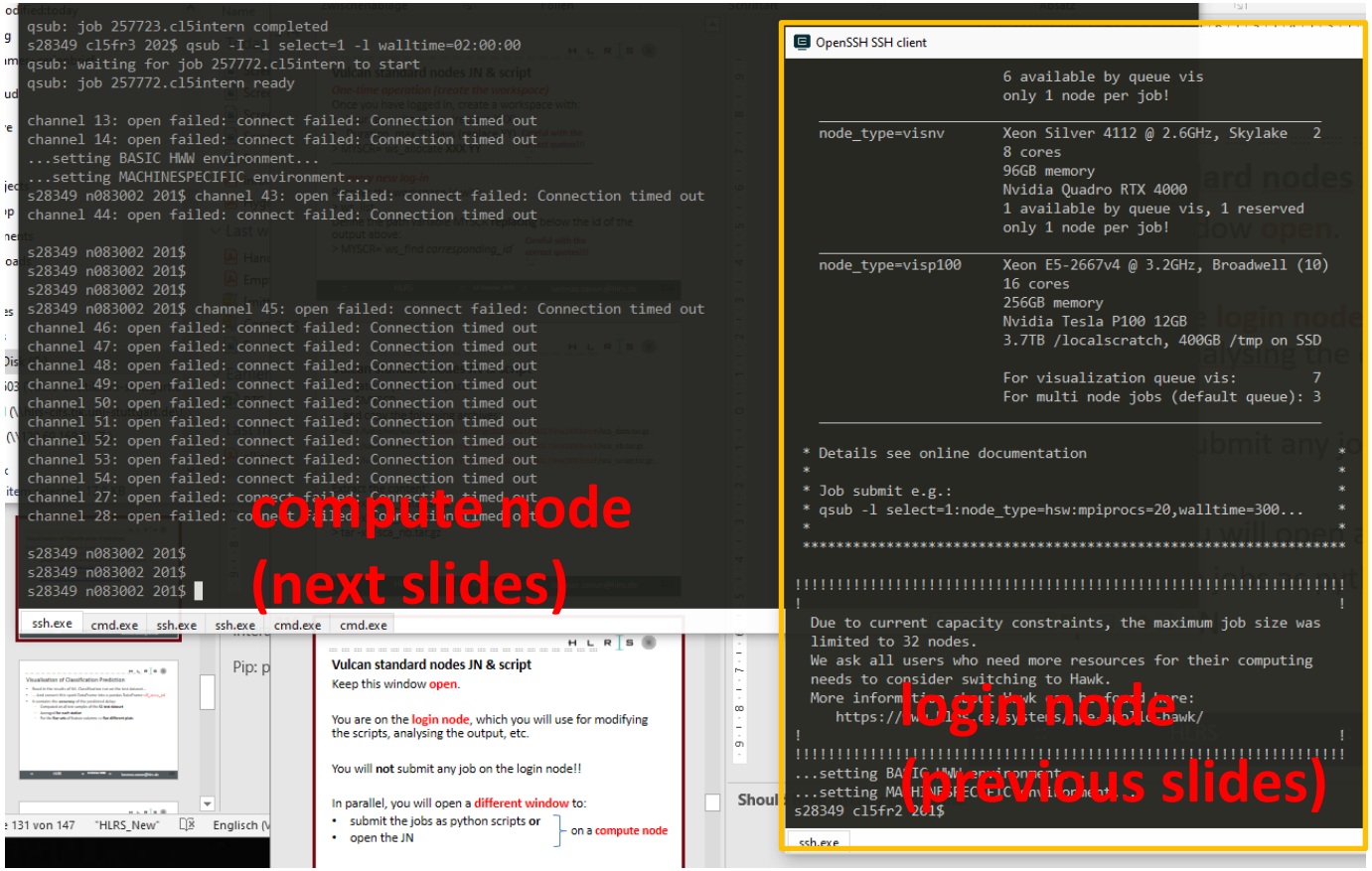
Start the cluster session

You can work with two different windows

- Login window:
  - **Stays open.**
  - You are on the **login node** for modifying the scripts, analysing the output, etc.
  - Do **not submit any job** on the login node!
- Compute node:
  - Open a **different window as in the next slides**.
  - Submit the jobs as python scripts **or**
  - open the Jupyter Notebook (with port forwarding)
    → then, this window will be unusable.

Start the cluster session

Start two terminals:



JN – BEFORE the exercise

Login **in a new window**:

> `ssh` *username*@training.hlrs.de

RAM

> `qsub -I -q smp -l select=1:node_type=skl:ncpus=`**`20`**`:mem=`**`80gb`**`,walltime=08:00:00`

These should all be "minus"!

You can enter another (lower) value `hh:mm:ss`

Wait **a few seconds** for availability.
Navigate to the workspace:

> **`cd $(ws_find`** *`yourWorkspace`*`)`

Initialise Spark and the Jupyter Notebook:

> **`. Notebooks/init_jn.sh`**

You should ALWAYS call `init` from the workspace home!

This should execute **immediately**. If it hangs, you might still be on the log-in node.

JN – BEFORE the exercise

DO NOT EXECUTE!!!
It is already included in `init_jn.sh`

`init_jn.sh` contains:

Module name might have changed!

```
pbsnodes -j $(cat $PBS_NODEFILE)
module load bigdata/spark_cluster/3.4.2
MYSCR=$(pwd)
cd $MYSCR
export MYWS=$MYSCR
init-spark
jupyter notebook --no-browser --ip=$(hostname)
```

Display the node's properties

Load the cluster

Set correct directory and export path

Initialise Spark and launch the JN.

JN – BEFORE the exercise

Copy from the output of the previous command the **URL** to the Jupyter Notebook.

```
resources_available.Qlist = compute
resources_available.switch = S-8f1050020008e
resources_available.vnode = n091602
resources_assigned.accelerator_memory = 0kb
resources_assigned.hbmem = 0kb
resources_assigned.mem = 0kb
resources_assigned.naccelerators = 0
resources_assigned.ncpus = 1
resources_assigned.nodecounter = 1
resources_assigned.vmem = 0kb
comment = hsw128gb20c
resv_enable = True
sharing = force_excl
last_state_change_time = Mon Oct 19 16:06:14 2020
last_used_time = Mon Oct 19 16:04:54 2020

ING: You have a single node job running in 'local master' mode. Daemons will not be started.
7:08:02.394 NotebookApp] Serving notebooks from local directory: /lustre/nec/ws2/ws/hpclzano-hsw_sbahn
7:08:02.394 NotebookApp] Jupyter Notebook 6.1.4 is running at:
7:08:02.394 NotebookApp] http://n091602:8888/?token=efd573af9fdfc82f42f07c608b4543fdcea86ff9e1e2f0db
7:08:02.394 NotebookApp]  or http://127.0.0.1:8888/?token=efd573af9fdfc82f42f07c608b4543fdcea86ff9e1e2f0db
7:08:02.394 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
7:08:02.403 NotebookApp]

To access the notebook, open this file in a browser:
    file:///zhome/academic/HLRS/hlrs/hpclzano/.local/share/jupyter/runtime/nbserver-7830-open.html
Or copy and paste one of these URLs:
    http://n091602:8888/?token=efd573af9fdfc82f42f07c608b4543fdcea86ff9e1e2f0db
 or http://127.0.0.1:8888/?token=efd573af9fdfc82f42f07c608b4543fdcea86ff9e1e2f0db
```

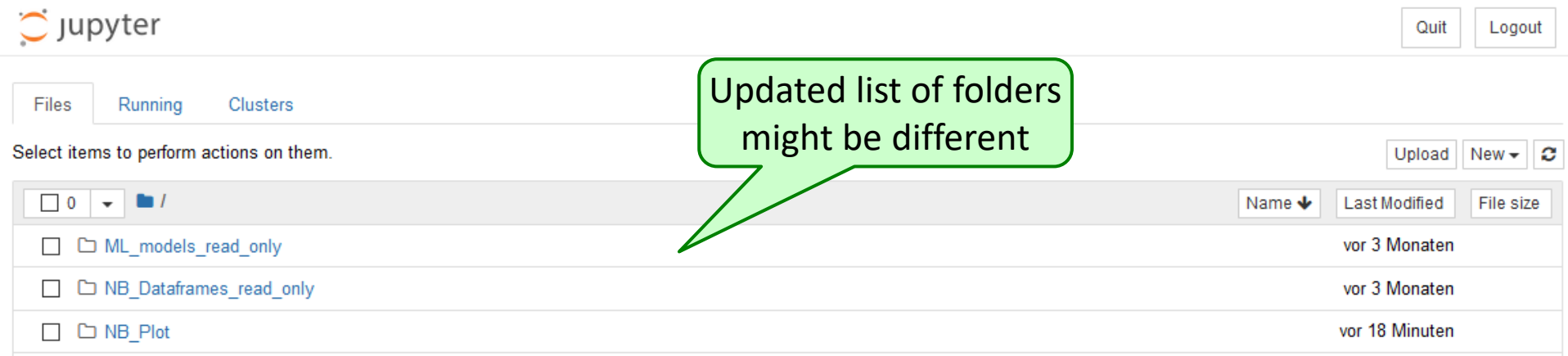
Copy THIS link: http://n…

JN – BEFORE the exercise

Jupyter-Chrome

Now start the Jupyter Notebook:

- Launch the **browser profile** as explained in the **login slides:** https://fs.hlrs.de/projects/par/events/2024/dl-hlrs/DL-HLRS-login.pdf
- Paste the **link** to the Notebook in the browser.



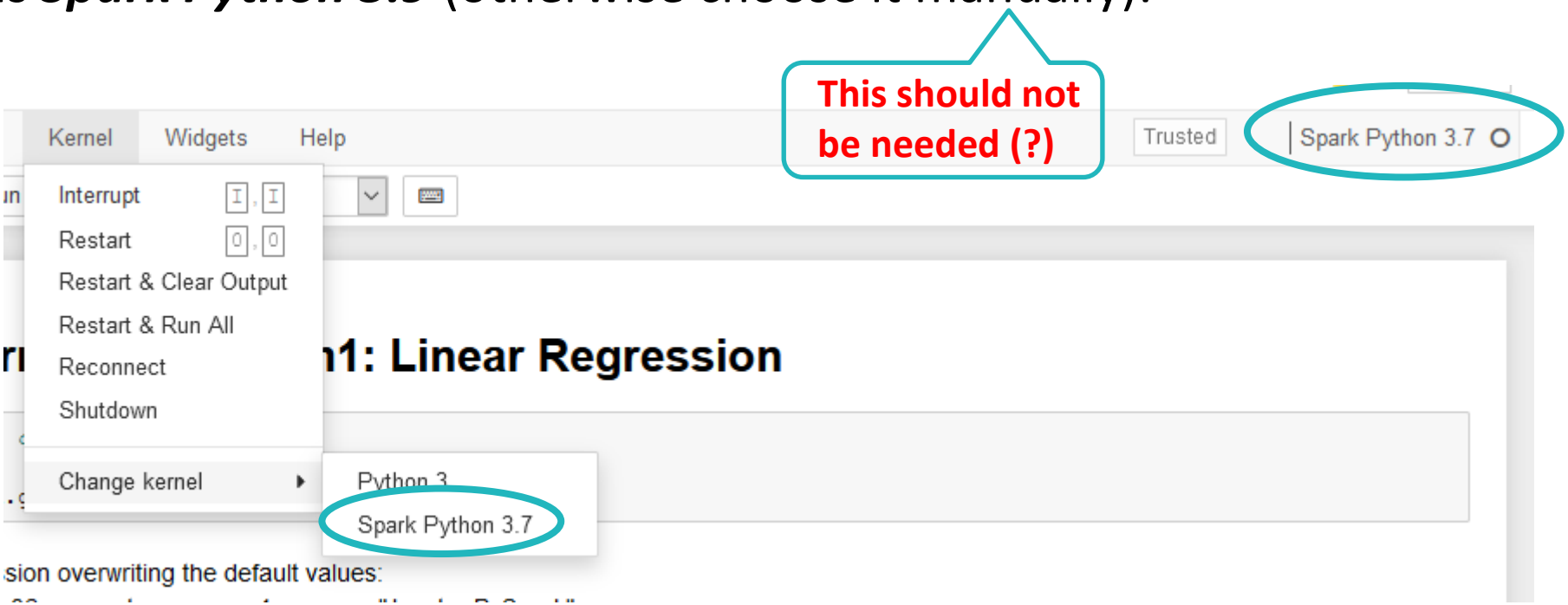
JN – BEFORE the exercise

Open the respective files in the folder `Notebooks` :

- Manipulation
  `NB1_manipulation.ipynb`
- Visualisation of manipulated data
  `NB2_vis-man.ipynb`
- Machine Learning: Linear Regression
  `NB3_linreg_ALL.ipynb`
- Machine Learning: Random Forest
  `NB4_class.ipynb`
- Visualisation of Machine Learning results
  `NB5_vis-class.ipynb`
- Handlers: Container of all user-defined functions
  `handlers.ipynb`

JN – BEFORE the exercise

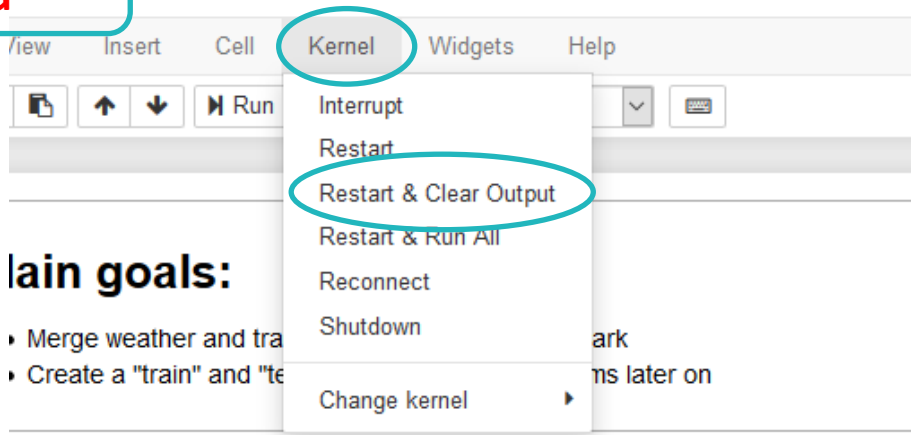
When opening any notebook, make sure that the kernel (top-right) is ***Spark Python 3.9*** (otherwise choose it manually):



JN – BEFORE the exercise

Choose: **Kernel → "Restart and clear output"** to start with a clean workspace.

This should not be needed



JN – DURING the exercise

If the Notebook crashes…

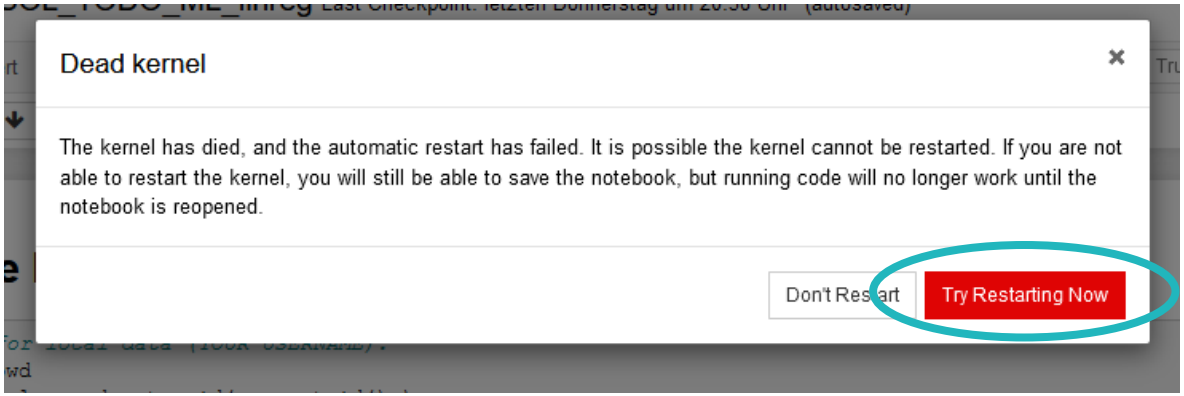
This should not be needed

In the terminal:

- Interrupt the kernel with **Ctrl+C** (and confirm [yes]),
- If the job is still running, repeat

```
> .   Notebooks/init_jn.sh
```

In the Notebook: Restart the kernel



JN – DURING the exercise

If the Notebook crashes…

If you are back on the **login node**:
Restart the procedure from the beginning (`qsub` etc.).

JN – DURING the exercise

Proceeding with the exercises, you will create some more folders that appear in the JN dashboard:

0 /	Name	Last Modified	File size
ML_models		vor 4 Tagen	
ML_models_read_only		vor 17 Tagen	
NB_ScriptPlot		vor 18 Tagen	
Notebooks		vor einer Stunde	
sbahn_data		vor 5 Tagen	
ScriptDataframes		vor 4 Stunden	
ScriptDataframes_read_only		vor 17 Tagen	
scripting		vor einem Tag	
ScriptPlot		vor 10 Tagen	

→ next slide

JN – DURING the exercise

→ Most folders have both a **Notebook** ~~and a **script**~~ **version**:

- **NB_ML_models**:
  ML models created by training the datasets.
- **ML_models_read_only**:
  ML models to read-in (pre-trained models).
- **NB_Dataframes**: Generated DataFrames.
- **NB_Dataframes_read_only**:
  Dataframes to read-in (pre-loaded Dataframes).
- **NB_Plot**: Folders with the generated plots.

JN – DURING the exercise

In each Notebook, e.g.
`Notebooks/NB1_manipulation.ipynb`

… replace only:

____ = exercise, replace! (you will get an error otherwise)

Solution, e.g.:
`Solutions/NB1-SOL_manipulation.ipynb`

Do one Notebook at a time!

JN – DURING the exercise

Unexecuted:

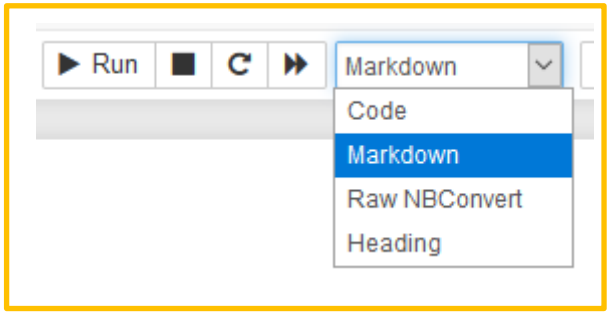
- Markdown cell
- Code cell

```
## Main goals:
* Merge weather and train data with Pandas and Spark

* Create a "training" and "test" data set for ML algo
```

```
In [ ]: %spark 16 32g Sbahn
```

Use **Shift+Return** (or ▶ Run) **to execute cells:**




```
In [*]: %spark 16 32g Sbahn
```

Executed:

- Markdown cell (**double click** to edit)
- Code cell

Main goals:

- Merge weather and train data with Pandas and Spark
- Create a "training" and "test" data set for ML algorithm

```
In [1]: %spark 16 32g Sbahn

Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.6
      /_/

Using Python version 3.7.9 (default, Sep 23 2020 17:09:01)
SparkSession available as 'spark'.
```

JN – DURING the exercise

Also useful:

Spark and Pandas reference pages to look up functions.

Spark:

https://spark.apache.org/docs/latest/api/python/

- Use “Search the docs“ up on the left.
- Needed solution usually follows: “pyspark.sql.“
  SQL = Spark module for structured data processing
  → DataFrames

JN – DURING the exercise

Also useful:

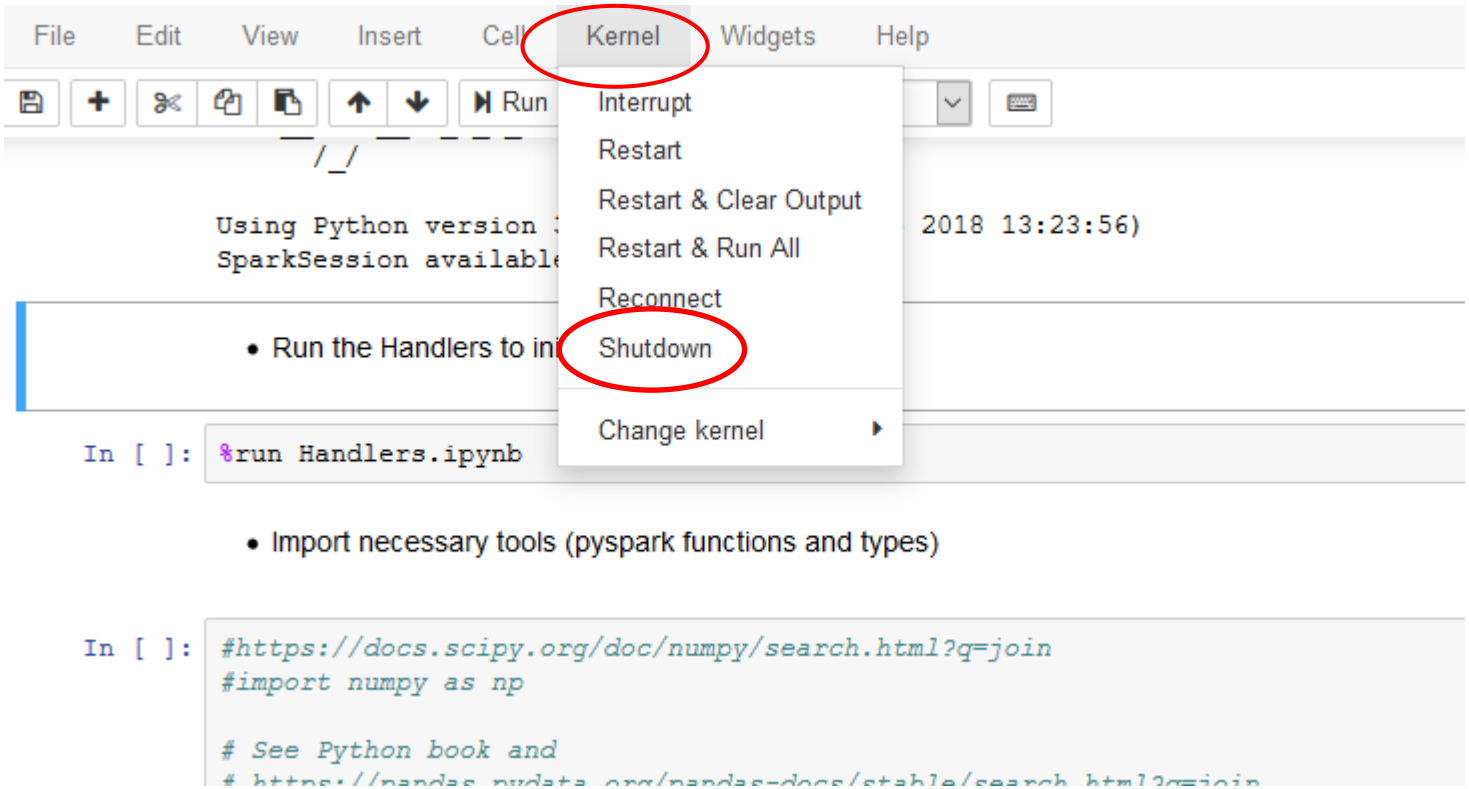
Pandas:

https://pandas.pydata.org/pandas-docs/stable/index.html

- Use “Search the docs“ up on the left.
- Needed solution usually follows: “pandas.DataFrame“
- Look in the Parameters list for the needed parameters.

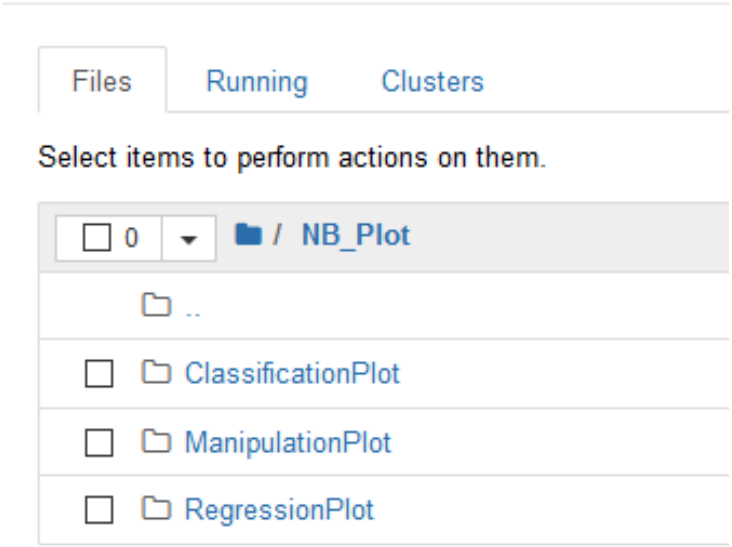
JN – AFTER the exercise

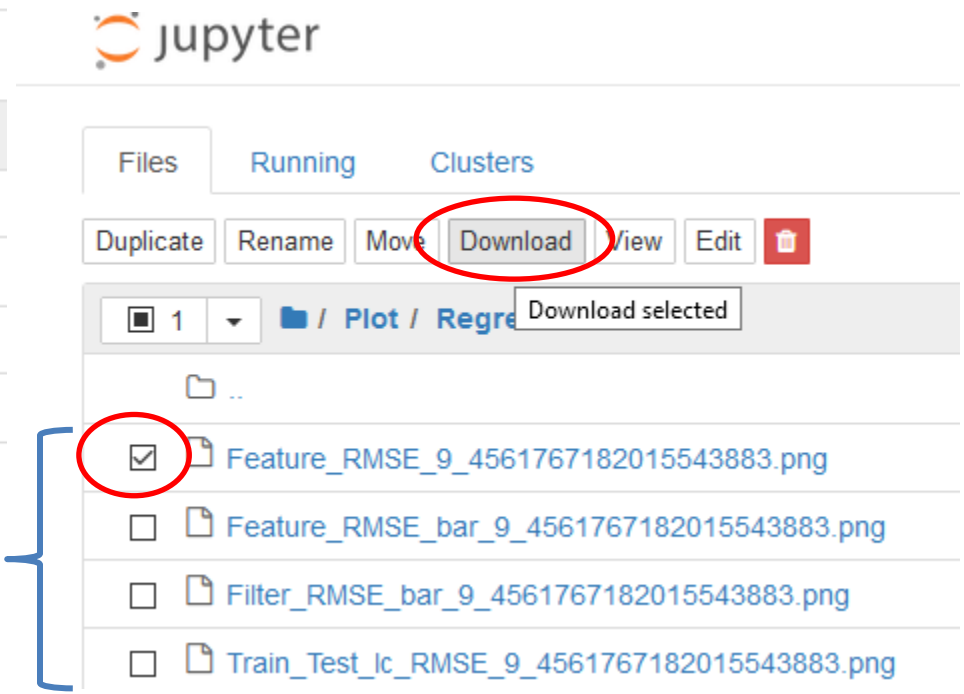
At the end of each Notebook, *shutdown* the kernel to free memory and clear all variables:



JN – AFTER the exercise

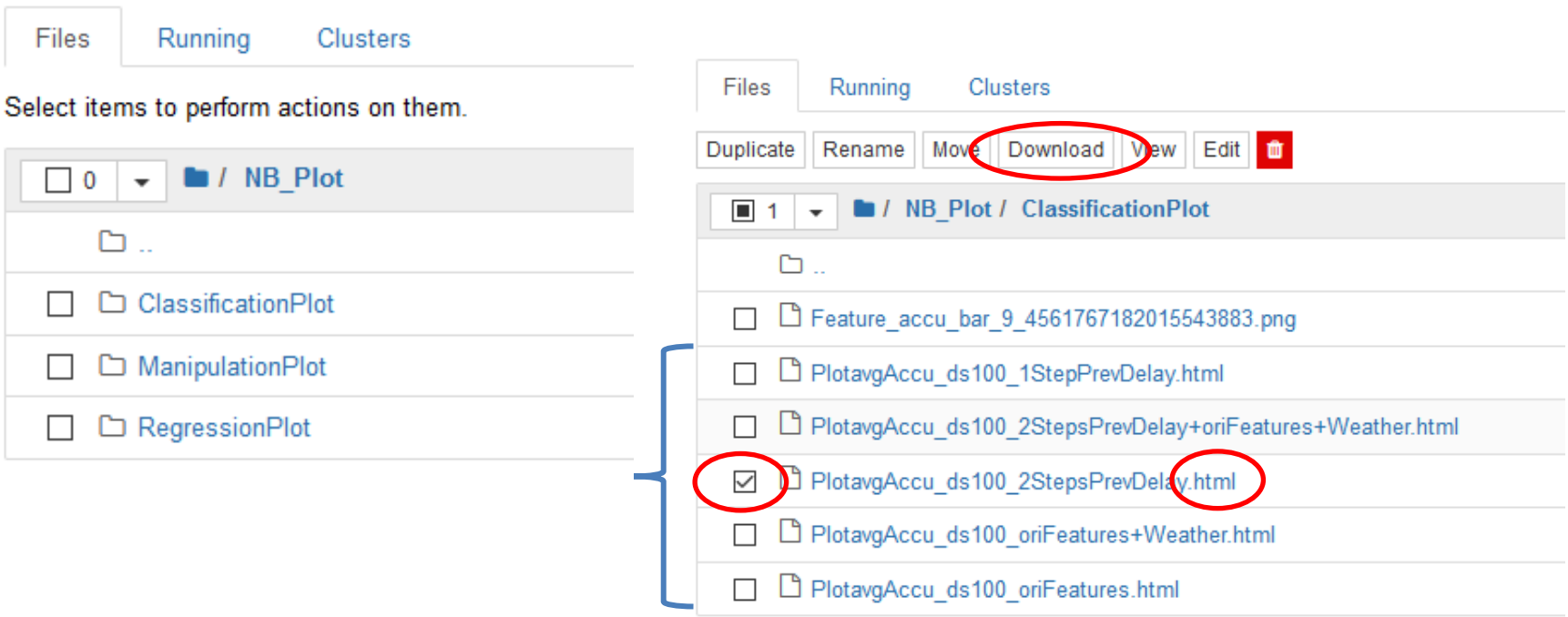
You can *download* any created plot in the plot folder and sub-folders:





JN – AFTER the exercise

Visualise the **html plots** (Notebook `NB5_vis-class.ipynb`) by downloading and opening them **locally** in a separate browser (**not** the JN browser profile!).



Material of the course

https://fs.hlrs.de/projects/par/events/2024/dl-hlrs

Notebooks: Download them directly, or

~~(accounts typically expire on Monday after the course):~~

- ~~Make sure your HWW VPN is switched on~~
- Get the path to your workspace, e.g. using:
  - **`ws_list`** : complete paths to all your workspaces
  - **`pwd`** : the current path.
- In a **new terminal**, replace as needed and type **in one line**:

```
> scp -r username@training.hlrs.de:/path/to/workspace/and/folder
      /path/to/local/destination
```

… and save the data e.g. on a USB stick.