

Start the cluster session

- 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

Two different windows: Probably not needed

JN – BEFORE the exercise

Login in a new window:

```
> ssh vulcan.hww.hlrs.de
```

```
> qsub -I -q R_sst -l select=1:node_type=hsw:mem=100gb,walltime=03:00:00
```

These should
all be “minus”!

One node and one
submission each!

RAM

Max. 18:00 –
current time!

Wait **a few seconds** for availability (prompt n...).

Navigate to the workspace:

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

Initialise Spark and the Jupyter Notebook:

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

You should be in 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!!! Since
already included in init_jn.sh

init_jn.sh contains:

Module
name
might
have
changed!

pbsnodes -j \$(cat \$PBS_NODEFILE)

Display the node's
properties

module load bigdata/spark_cluster/3.3.1

Load the cluster

MYSCR=\$(pwd)

cd \$MYSCR

export MYWS=\$MYSCR

Set correct
directory and
export path

Initialise Spark and
launch the JN.

init-spark

jupyter notebook --no-browser --ip=\$(hostname)

JN – BEFORE the exercise

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

```
resources_available.nodename = n091602
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

INFO: 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=efd573af9fd82f42f07c608b4543fdcea86ff9e1e2f0db
7:08:02.394 NotebookApp] or http://127.0.0.1:8888/?token=efd573af9fd82f42f07c608b4543fdcea86ff9e1e2f0db
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=efd573af9fd82f42f07c608b4543fdcea86ff9e1e2f0db
  or http://127.0.0.1:8888/?token=efd573af9fd82f42f07c608b4543fdcea86ff9e1e2f0db
```

Copy THIS link:
[http://n...](http://n091602:8888/?token=efd573af9fd82f42f07c608b4543fdcea86ff9e1e2f0db)

JN – BEFORE the exercise

Now start the Jupyter Notebook:

- Launch the prepared browser on desktop
- Paste the **link** to the Notebook in the browser.
- Launch Jupyter-Chrome **only once**.



Jupyter-Chrome

 jupyter

Quit

Logout

Files Running Clusters

Select items to perform actions on them.

Upload

New ▾



<input type="checkbox"/> 0 ▾	 /	Name ▾	Last Modified	File size
<input type="checkbox"/>	ML_models_read_only		vor 3 Monaten	
<input type="checkbox"/>	NB_Dataframes_read_only		vor 3 Monaten	
<input type="checkbox"/>	NB_Plot		vor 18 Minuten	

Updated list of folders
might be different

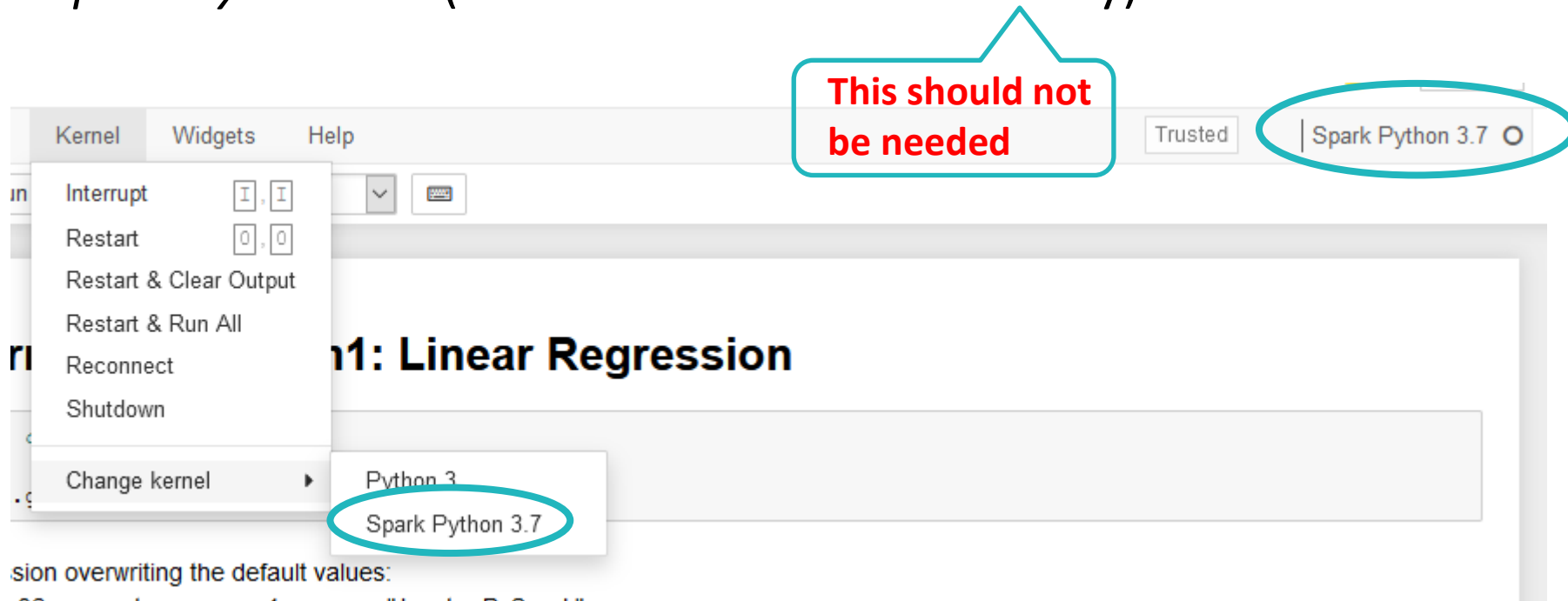
JN – BEFORE the exercise

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

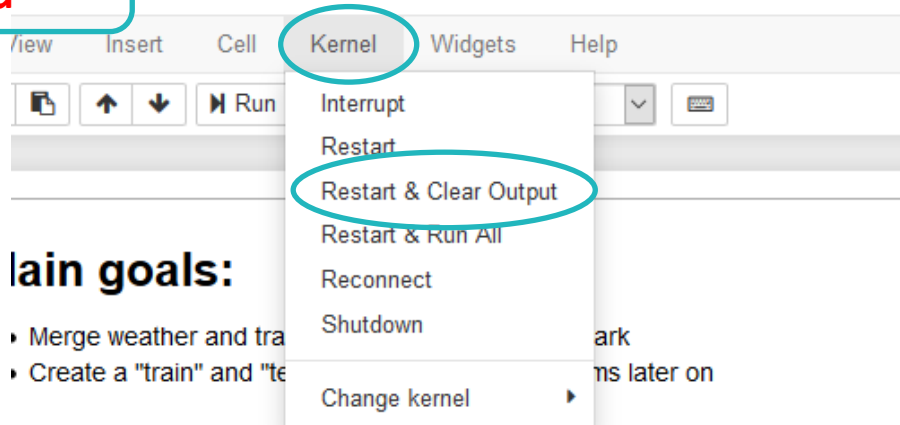
When opening any notebook, make sure that the kernel (top-right) is *Spark Python 3.7* (otherwise choose it manually):



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

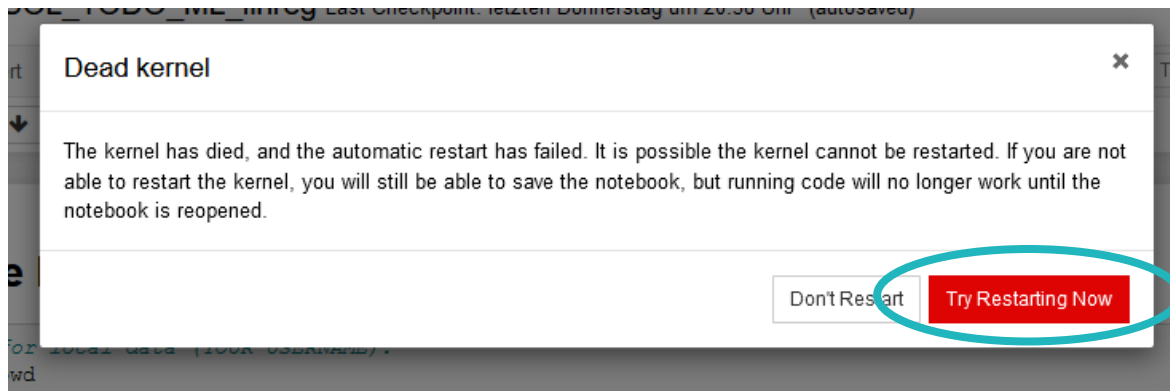
If the Notebook crashes...

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 will appear in the JN dashboard:

<input type="checkbox"/>	0		Name ↓	Last Modified	File size
<input type="checkbox"/>		ML_models		vor 4 Tagen	
<input type="checkbox"/>		ML_models_read_only		vor 17 Tagen	
<input type="checkbox"/>		NB_ScriptPlot		vor 18 Tagen	
<input type="checkbox"/>		Notebooks		vor einer Stunde	
<input type="checkbox"/>		sbahn_data		vor 5 Tagen	
<input type="checkbox"/>		ScriptDataframes		vor 4 Stunden	
<input type="checkbox"/>		ScriptDataframes_read_only		vor 17 Tagen	
<input type="checkbox"/>		scripting		vor einem Tag	
<input type="checkbox"/>		ScriptPlot		vor 10 Tagen	

→ next slide

JN – DURING the exercise

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

- **NB_ML_models** and **Script_ML_models**: ML models created by training the datasets.
- **ML_models_read_only**: ML models to read-in (pre-trained models).
- **NB_** and **Script_Dataframes**: Generated DataFrames.
- **NB_** and **Script_Dataframes_read_only**: Dataframes to read-in (pre-loaded Dataframes).
- **NB_** and **Script_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.: NB1-**SOL**_DataManipulation.ipynb

Do one Notebook at a time!

JN – DURING the exercise

Unexecuted:

- Markdown cell
- Code cell

The screenshot shows a notebook cell with a light gray background. At the top, there is a blue heading `## Main goals:`. Below it are two bullet points in blue text: `* Merge weather and train data with Pandas and Spark` and `* Create a "training" and "test" data set for ML algo`. Below the cell, the input area shows `In []: %spark 16 32g Sbahn`.

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

The screenshot shows the top toolbar of a notebook cell. The 'Run' button (a play icon) is highlighted with a yellow box. A dropdown menu is open, showing options: 'Markdown', 'Code', 'Markdown', 'Raw NBConvert', and 'Heading'. The first 'Markdown' option is selected.

Executed:

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

The screenshot shows the notebook cell after execution. The input area now shows `In [*]: %spark 16 32g Sbahn`. The output area contains the following text: `Welcome to`, a large stylized logo for Databricks, and `version 2.4.6`. Below the logo, it says `Using Python version 3.7.9 (default, Sep 23 2020 17:09:01)` and `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

SS Details skipped

JN – DURING the exercise

Also useful:

Pandas:

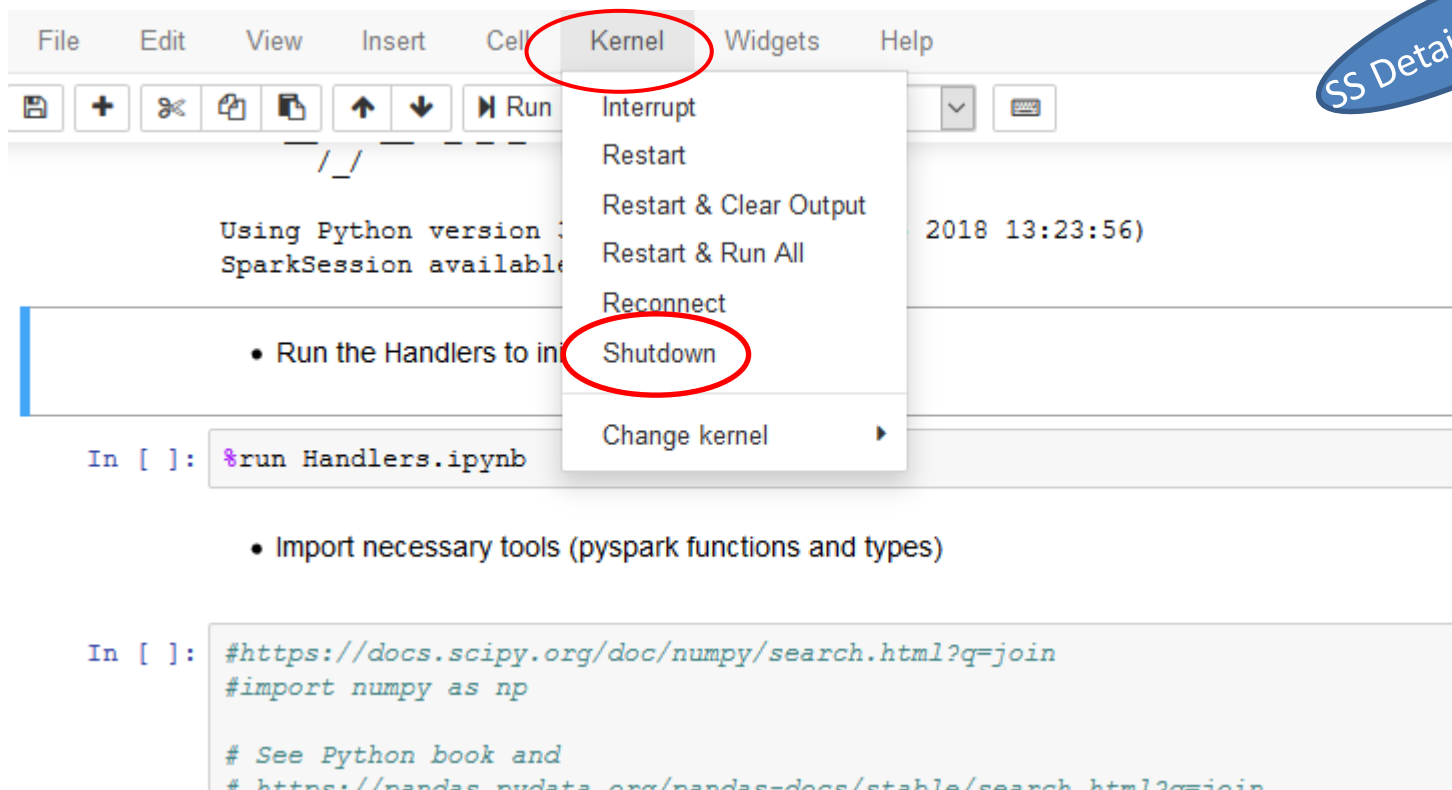
<https://pandas.pydata.org/pandas-docs/stable/index.html>

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

SS Details skipped

JN – AFTER the exercise

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



The screenshot shows the Jupyter Notebook interface. The 'Kernel' menu is open, and the 'Shutdown' option is highlighted with a red circle. A blue oval with the text 'SS Details skipped' is overlaid on the right side of the interface. The notebook content includes a code cell with the following text:

```
In [ ]: %run Handlers.ipynb
```

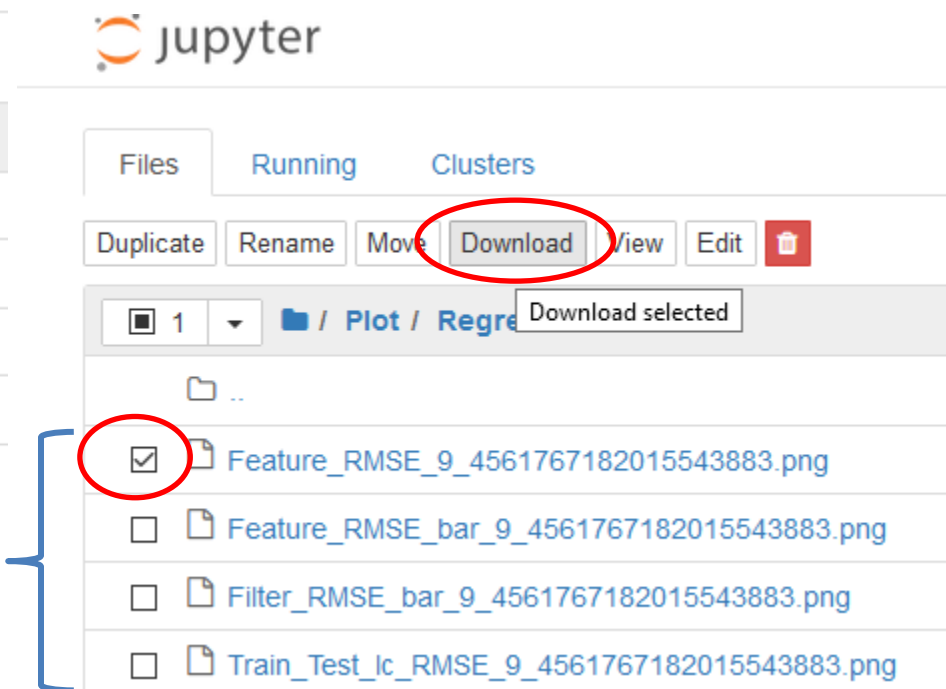
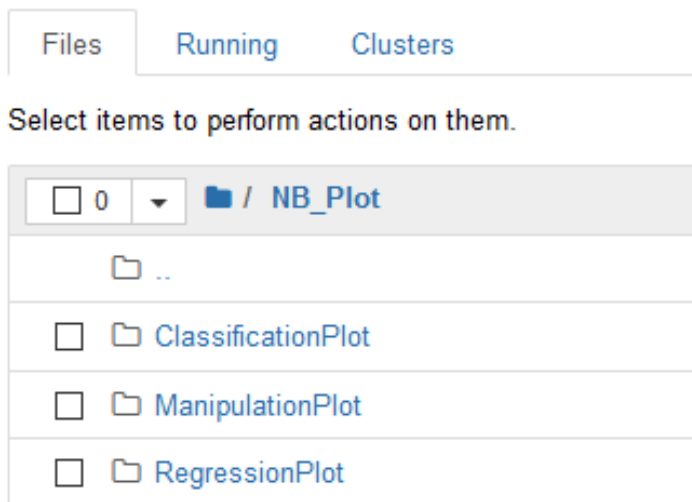
- Run the Handlers to in

```
In [ ]: #https://docs.scipy.org/doc/numpy/search.html?q=join
#import numpy as np

# See Python book and
# https://pandas.pydata.org/pandas-docs/stable/search.html?q=join
```

JN – AFTER the exercise

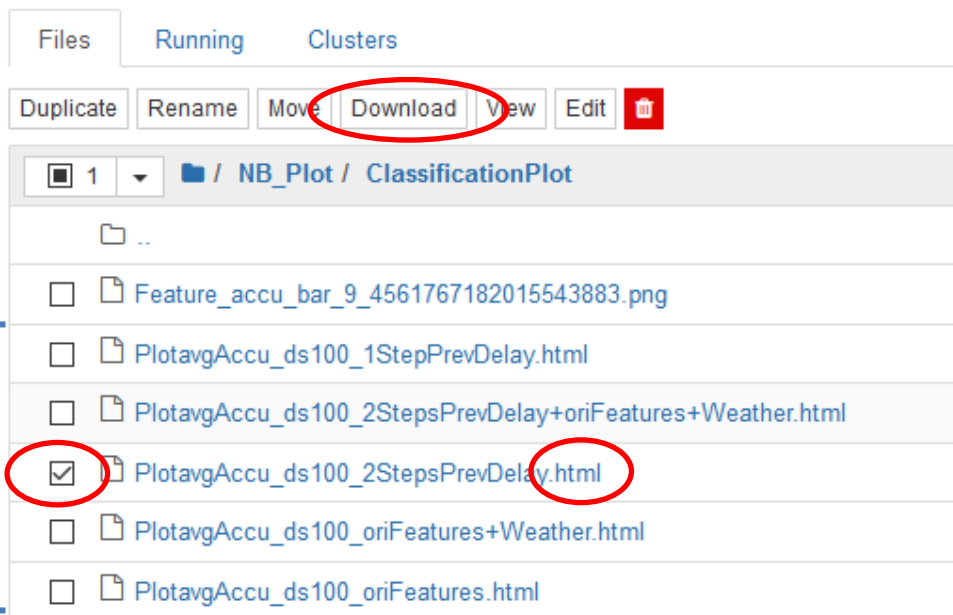
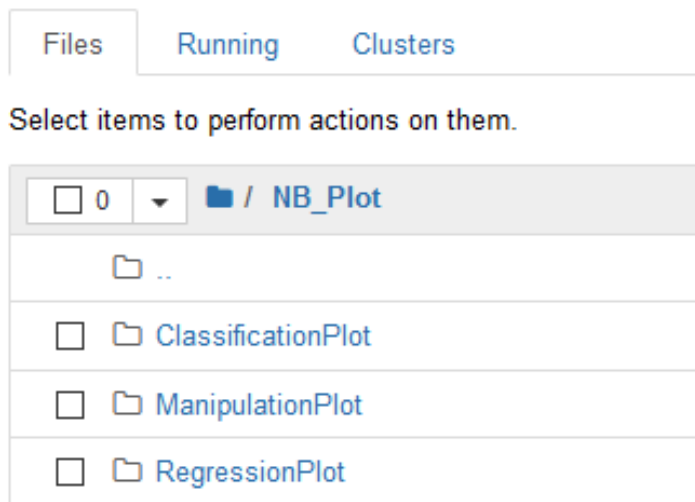
You can *download* any modified notebook or created plot:



JN – AFTER the exercise

Visualise the **html plots** (Notebook NB5_vis_class) by downloading and opening them **locally** in a separate browser (**not** the JN browser profile!).

SS Details skipped



Material of the course

<https://fs.hlrs.de/projects/par/events/2023/sst>

Notebooks: Download them directly, or:

- In **Vulcan**, get the path to your workspace:
 - **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@vulcan.hww.hlrs.de:/path/to/workspace/and/folder  
/path/to/local/destination
```

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