

Create your workspaces

- Follow the next slides to copy data and scripts of the exercises to your workspace.
- Please be careful, since copy-pasting from pdf might be inaccurate.

- **To navigate to a workspace**

get the workspace id with `ws_list`:

```
> ws_list:
```

to define the path variable `MYSCR`:

```
> MYSCR=$(ws_find wspart3)
```

Navigate to your workspace:

```
> cd $MYSCR
```

Before the exercise

- Login in a new terminal(cmd, powershell, etc.)
 - ssh username@vulcan.hww.hlr.de
 - qsub -I -q R_sst -l select=1:node_type=hsw:mem=100gb,walltime=03:00:00

- Wait a few seconds for availability, then navigate to your workspace:

```
cd $(ws_find workspace)
```

- Initialise the Jupyter Notebook:

```
. module load python  
. jupyter notebook --no-browser --ip=$(hostname)
```

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

Before the exercise

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

```
available extensions :
s34361 n070901 201$ cd /lustre/nec/ws3/ws/hpclaali-ss23
s34361 n070901 202$ module load bigdata/spark_cluster/3.3.1
s34361 n070901 203$ MYSCR=$(pwd)
s34361 n070901 204$ cd $MYSCR
s34361 n070901 205$ export MYWS=$MYSCR
s34361 n070901 206$ jupyter notebook --no-browser --ip=`hostname`
[I 09:16:26.587 NotebookApp] Serving notebooks from local directory: /lustre/nec/ws3/ws/hpclaali-ss23
[I 09:16:26.587 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 09:16:26.587 NotebookApp] http://n070901:8888/?token=1eefbe4d1ab73b4b5d080f707681f94dd463768e2a791b0d
[I 09:16:26.587 NotebookApp] or http://127.0.0.1:8888/?token=1eefbe4d1ab73b4b5d080f707681f94dd463768e2a791b0d
[I 09:16:26.587 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confi
[C 09:16:26.596 NotebookApp]

To access the notebook, open this file in a browser:
  file:///zhome/academic/HLRS/hlrs/hpclaali/.local/share/jupyter/runtime/nbserver-77433-open.html
Or copy and paste one of these URLs:
  http://n070901:8888/?token=1eefbe4d1ab73b4b5d080f707681f94dd463768e2a791b0d
  or http://127.0.0.1:8888/?token=1eefbe4d1ab73b4b5d080f707681f94dd463768e2a791b0d
```

Copy this URL

Before the exercise

Now start the Jupyter Notebook:

- Launch the prepared browser on desktop.
- Paste the URL of Jupyter notebook in the browser.
- Open the next jupyter notebook file:
[Transformer.ipynb](#)

Select items to perform actions on them.

Upload New ↕

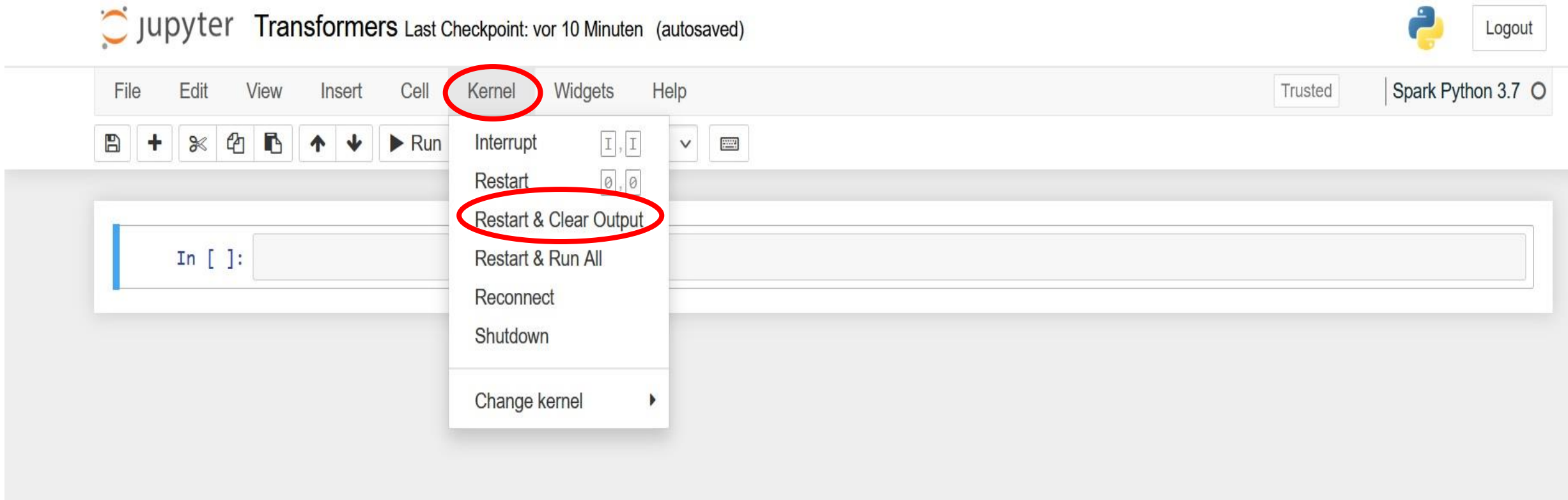
<input type="checkbox"/> 0	▼	📁 /	Name ↓	Last Modified	File size
<input type="checkbox"/>	📁	Distilbert_based_transformer		vor 4 Stunden	
<input type="checkbox"/>	📁	distilbert_model_uncased		vor einem Tag	
<input type="checkbox"/>	📁	gpt2_pretrained		vor 2 Stunden	
<input type="checkbox"/>	📁	my_transformer		vor 2 Stunden	
<input type="checkbox"/>	📁	tokenizer		vor einem Tag	
<input type="checkbox"/>	📁	tokenizer_pt		vor 6 Stunden	
<input type="checkbox"/>	📄	Transformer.ipynb	Running	vor 11 Minuten	45.4 kB
<input type="checkbox"/>	📄	ecomm_data.csv		vor einem Tag	36.9 MB



Open this file

During the exercise

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



The screenshot shows the JupyterLab interface for a notebook named "Transformers". The top bar includes the Jupyter logo, the notebook name, and a status message: "Last Checkpoint: vor 10 Minuten (autosaved)". On the right, there is a Python logo and a "Logout" button. Below the top bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". The "Kernel" menu is open, showing options: "Interrupt", "Restart", "Restart & Clear Output", "Restart & Run All", "Reconnect", "Shutdown", and "Change kernel". The "Restart & Clear Output" option is circled in red. Below the menu bar is a toolbar with icons for saving, adding, deleting, copying, pasting, undo, redo, and running. The main area shows a code cell with "In []:" and an empty input field.

During the exercise

- You can download the pretrained model's files.

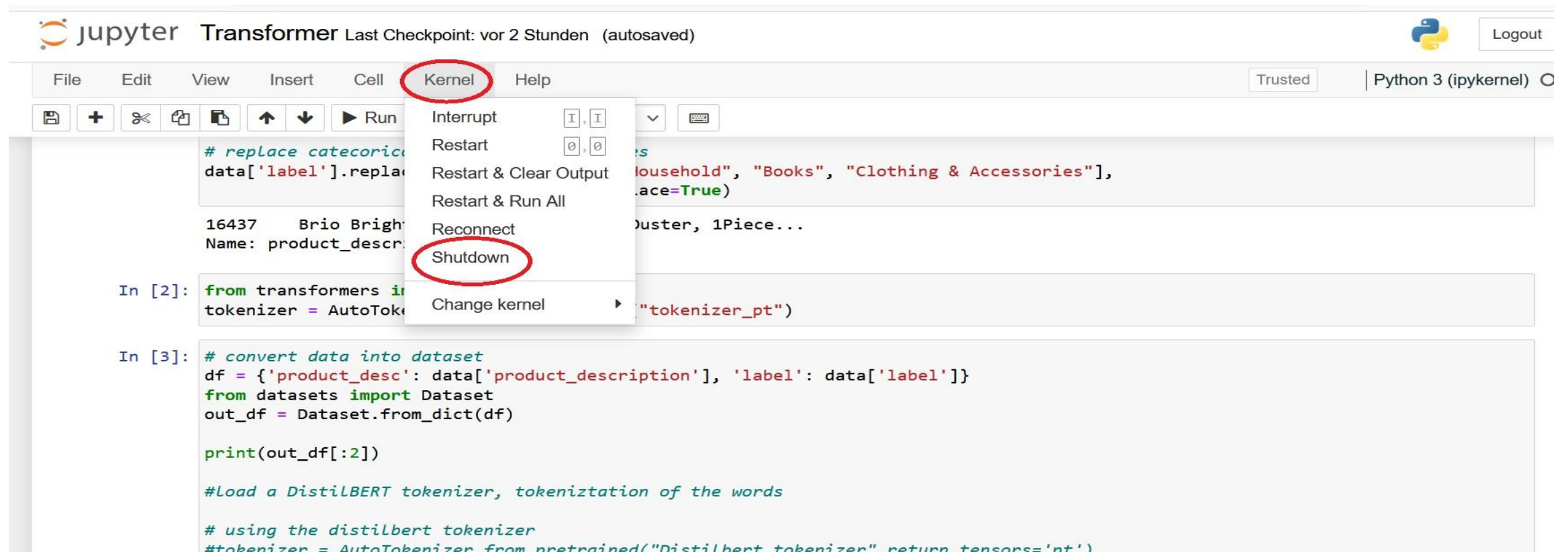


The screenshot shows a web browser window with a JupyterLab interface. The browser address bar shows the URL `n121201:8888/tree/distilbert_model_uncased`. The JupyterLab interface has a top bar with the 'jupyter' logo and 'Quit' and 'Logout' buttons. Below this is a navigation bar with 'Files', 'Running', and 'Clusters' tabs. The 'Files' tab is active, showing a file browser for the directory `distilbert_model_uncased`. The file browser has a toolbar with buttons for 'Duplicate', 'Move', 'Download', 'View', 'Edit', and a trash icon. The 'Download' button is circled in red. Below the toolbar is a table of files:

	Name	Last Modified	File size
<input checked="" type="checkbox"/>	..	vor ein paar Sekunden	
<input checked="" type="checkbox"/>	config.json	vor einem Tag	832 B
<input checked="" type="checkbox"/>	pytorch_model.bin	vor einem Tag	268 MB

After the exercise

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



The screenshot shows a Jupyter Notebook interface. The top bar displays the Jupyter logo, the notebook title "Transformer", and the last checkpoint information "Last Checkpoint: vor 2 Stunden (autosaved)". On the right, there is a Python logo and a "Logout" button. The main menu bar includes "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". The "Kernel" menu is open, showing options: "Interrupt", "Restart", "Restart & Clear Output", "Restart & Run All", "Reconnect", "Shutdown", and "Change kernel". The "Shutdown" option is circled in red. Below the menu, the notebook content is visible, showing code cells. The first cell contains a comment "# replace categories" and a line of code: `data['label'].replace("household", "Books", "Clothing & Accessories"),`. The second cell, labeled "In [2]:", contains code for loading a DistilBERT tokenizer: `from transformers import AutoTokenizer; tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")`. The third cell, labeled "In [3]:", contains code for creating a dataset and printing it: `df = {'product_desc': data['product_description'], 'label': data['label']}; from datasets import Dataset; out_df = Dataset.from_dict(df); print(out_df[:2])`. Below this, there are more comments and code for loading the tokenizer: `#Load a DistilBERT tokenizer, tokenization of the words; # using the distilbert tokenizer; #tokenizer = AutoTokenizer.from_pretrained("Distilbert tokenizer" return tensors='nt')`.

Material of the course

- <https://fs.hlrs.de/projects/par/events/2023/sst>
- To have a local copy of the Notebooks:
 - In a local terminal, replace the below command with your account, remote and local destination:
 - `> scp -r username@vulcan.hww.hlrs.de:/path/to/workspace/and/folder /path/to/local/destination`
- To get the needed paths, type on a login node in Vulcan:
 - `ws_list` for the complete paths to all your workspaces.
 - `pwd` for the current path.

At the End of the exercise

H L R I S

- Please log out from your laptop. Thank you!