

*Prof. Dr. Jochen Garcke  
Dr. Bastian Bohn  
Paolo Climaco  
Arno Feiden  
Sara Hahner*



## TECHNICAL INTRODUCTION

---

Welcome to our practical lab „Algorithms in Machine Learning and Their Application“, where we equip you with the necessary skills and tools to write numerical software for machine learning and apply machine learning for data analysis task.

This sheet provides installation guidelines for the necessary tools as well as exercises to make you familiar with the workflow employed in this lecture.

If you encounter any problems during the installation process or in the exercises, please use the eCampus forum, or write an e-mail, and ask for help.

Please note that you also have the possibility to get access to computer resources at the INS.

### *The Terminal*

The terminal is a program that allows users to interact with their operating system's services. This is especially useful when running programs without graphical user interfaces (e.g. your own) and therefore the terminal is an invaluable tool for programmers. It is also called shell, command line or console. Strictly speaking, these terms are names for different parts and aspects of the program, but they are commonly used interchangeably.

You should invest some time to get used to handling the command line efficiently. The sources below should help you to get familiar with the most important commands of your preferred platform.

- Unix-like systems (including Linux and macOS):
  - [https://www.youtube.com/watch?v=3\\_vgwWT01Vk](https://www.youtube.com/watch?v=3_vgwWT01Vk)
  - [https://en.wikipedia.org/wiki/List\\_of\\_Unix\\_commands](https://en.wikipedia.org/wiki/List_of_Unix_commands)
- Windows:
  - <https://www.youtube.com/watch?v=TUNNmVeyjW0>
  - <https://programminghistorian.org/en/lessons/intro-to-powershell>
  - <https://devblogs.microsoft.com/scripting/table-of-basic-powershell-commands/>

**Task 0.1** (Terminal). *Make yourself familiar with your terminal. Make sure you know how to*

1. *create a new directory,*
2. *change directory (change into child and parent directories),*
3. *list all files in a directory,*
4. *remove single files and remove all files in a directory,*
5. *run a program*

*from your command line.*

### *Text Editors*

Source files and scripts are composed and edited using a text editor.

We specifically recommend to use VS Code. It can be downloaded for all platforms on their website <https://code.visualstudio.com/>, where you can also find installation guides and documentation. You are allowed to use any other text editor, but we will not offer editor specific support for any editor other than VS Code.

One of VS Code's strong suits is the vast number of different extensions that are available and which help to adapt the text editor program to your specific needs and tastes. You can download language extensions for most programming languages, spell checking extensions and a lot more directly in VS Code itself.

**Task 0.2** (Text editors). *Give VS Code at least a try. Make yourself familiar with your text editor and its features.*

### *Git and GitHub*

Git is a version control system that tracks changes in source code and simplifies software development in teams. Many software companies use it to organize their workflow and keep track of development progress.

There are several websites and companies that provide hosting for version control using Git. In this practical lab we use GitHub (<https://github.com/>).

We use Git and GitHub classroom (<https://classroom.github.com/>) to hand in exercises, automatically test your solutions (and thus give you feedback on your programs) and facilitate the work in groups.

### *Install Git on your computer*

- **Linux:** Open a terminal and install Git using your package manager. For Ubuntu, this is done by running

```
sudo apt-get update
sudo apt-get install git
```

For other Linux distributions, choose your corresponding package manager.

- **Windows:** We will use chocolatey to install most software on Windows.

To install chocolatey, open an elevated (Administrator) PowerShell and run

```
Set-ExecutionPolicy Bypass -Scope
↳ Process -Force; [System.Net.
↳ ServicePointManager]::
↳ SecurityProtocol = [System.
↳ Net.ServicePointManager]::
↳ SecurityProtocol -bor 3072;
↳ iex ((New-Object System.Net.
↳ WebClient).DownloadString('
↳ https://chocolatey.org/
↳ install.ps1'))
```

More information (including the command line above) can be found at

<https://chocolatey.org/install>.

After chocolatey has been installed successfully you can use

```
choco install git -y
```

to install Git.

- **macOS:** We will use homebrew to install most software on macOS.

To install homebrew, open a terminal and run

```
xcode-select --install
/bin/bash -c "$(curl -fsSL https://
↳ raw.githubusercontent.com/
↳ Homebrew/install/master/
↳ install.sh)"
```

More information (including the command above) can be found at

<https://brew.sh>.

After homebrew has been installed successfully you can use

```
brew install git
```

to install Git.

*Get familiar with GitHub*

Here are some tutorials for using Git:

- <https://www.youtube.com/watch?v=50LUolv4DaM&feature=youtu.be> (a well-structured demonstration of the Git workflow under Linux)
- <https://rogerdudler.github.io/git-guide/index.de.html>
- <https://guides.github.com/activities/hello-world/>

**Task 0.3** (Start using Git and GitHub).

1. *Install Git on your computer*
2. *Get an account on GitHub (<https://github.com/join?source=header-home>) and watch or read the tutorials above.*
3. *Get an account on GitHub Classroom (<https://classroom.github.com/>)*

*Remote Team Work*

When coding together in a team, it is not always possible (as during a pandemic) to be physically present in the same room. Fortunately, there are tools that enable programmers to code together remotely. One is, of course, the aforementioned version control system Git. Video calls and chat programs help to organize your team work.

However, sometimes you still need to code on the same computer. If you want to enable your group to code together on your computer, you can do so, for example, by using the Live Share extension of VS Code:

1. Download the Live Share extension of VS Code
2. Open your project in VS Code and click on the Live Share extension button.
3. Start a collaboration session and sign in with your GitHub account.
4. Invite your team or collaborator(s) by sending them the invitation link via e-mail (they need VS Code and the Live Share extension as well).
5. You will now be able to code in the same project independently or follow each other while you code.

*Python Interpreter*

The programming language of this lecture is Python 3 (<https://www.python.org/>). To run Python programs on your computer, you need a so-called Python interpreter (see instruction guidelines below). Python can be used both interactively in the interpreter as well as by executing Python scripts, i.e. text files with the ending `.py`. Note that your hand-in exercises will need to be Python scripts.

For Python beginners and those who are not confident with installing packages, we recommend to use the Anaconda distribution (<https://www.anaconda.com/products/individual>). It is easy to install on common operating systems and comes with all required packages preinstalled (and many, many more packages).

To install Python 3 on your computer, follow these instructions:

- **Linux:** Open a terminal and run

```
sudo apt-get update
sudo apt-get install python3
```

for Ubuntu. For other Linux distributions, choose your corresponding package manager.

- **Windows:** Open an elevated (Administrator) PowerShell and run

```
choco install python -y
```

to install Python 3.

- **macOS:** Open a terminal and run

```
brew install python
```

to install Python 3.

**Task 0.4** (Python interpreter). *Install an Python 3 interpreter on your computer and check that it installed successfully by running*

```
python3 --version
```

The following Python packages are required: numpy, matplotlib und jupyter. If you already have a Python installation, then you can install the packages via a package manager (for instance, pip or conda).

*Test Lab*

To get you used to the workflow of this practical lab, here is a small test classroom assignment.

**Task 0.5** (Test classroom assignment). 1. *Accept the assignment on GitHub classroom from <https://classroom.github.com/a/XXxJNnyX>. This will create a repository for your submissions that only you and the assistants of the course have access to.*

2. *Clone this repository to your local computer.*
3. *Start a new branch to work on.*
4. *Solve the exercise according to the information in the readme of the repository.*
5. *Stage, commit and push your changes.*
6. *Create a pull request on GitHub.*
7. *Repeat, if necessary until all tests pass.*

To learn more about git, you can also accept the assignment for the introduction to GitHub [https://classroom.github.com/a/\\_5rAtHaQ](https://classroom.github.com/a/_5rAtHaQ).