

# A tool a day keeps the bad review away

---

Pavol Harar<sup>1,2</sup>  0000-0001-5206-1794

ESI Summer School: Machine Learning for Materials Hard and Soft  
Vienna, 11. – 22. July 2022

<sup>1</sup>University of Vienna, <sup>2</sup>Brno University of Technology

During the life cycle of our machine learning research projects, we interact with source code, data, and models almost daily. Therefore, it is wise to make use of various tools and tricks of the community that

- 1) save a lot of time during the development (versioning),
- 2) can be decisive during the peer review (reproducibility), and
- 3) are invaluable for making an impact (presentation).

In this hands-on session, we will use those tools while developing a specific ML project. After the session, you should be able to use the tools in projects of your own.

Environment

Versioning

Presentation

Compute & Deploy

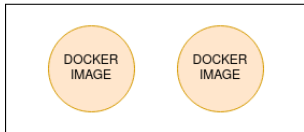
# Environment

---

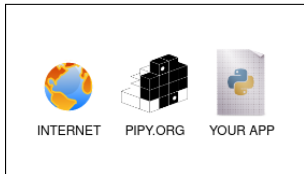


- powerful open source containerization platform
- rapid and continuous deployment of single-purpose machines
- isolates applications in a specified environment
- provides OS isolation, portability and security
- 2021 #1 most wanted and #2 most loved developer tool

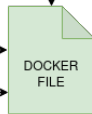
## DOCKER REGISTRY



## OTHER RESOURCES



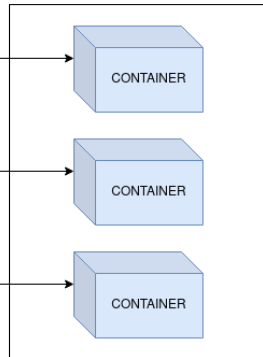
CREATE



BUILD



## DOCKER DEAMON ON HOST STYTEM



```
# installing docker.io (package by Debian) on Ubuntu
# check stackoverflow.com/questions/45023363#57678382
# on why you want to do that & what are the trade-offs
sudo apt install docker.io
# for other OS https://docs.docker.com/engine/install/
```

```
# useful commands
```

```
docker ps -a # list all containers
docker pull <image_name> # pull an image from registry
docker images # list all images
docker build <arguments> # build a new image
docker run <arguments> # run a container
docker rm <container id> # remove stopped container
docker rmi <image_name> # remove not used image
docker attach <container id> # connect to container
```



- Isolated Python environments
- Addresses the problem of dependencies, versions, and permissions
- Prevents mixing global with project dependencies
- Recreates the whole environment with a few commands



```
sudo pip3 install virtualenv
virtualenv .venv # create a folder for the env
source .venv/bin/activate # activate the env
pip install -r requirements.txt # install all deps
... # do your work
deactivate
```

```
# useful commands
pip --version
pip list # list all installed packages
pip show <packagename> # show package info
pip freeze # output all packages with exact versions
pip install <packagename>
pip install <packagename> --upgrade
pip uninstall <packagename>
```



- Package management and environment management system
- Runs on Windows, macOS, Linux and z/OS
- Easily creates, saves, loads and switches between environments

```
# download the miniconda .sh installer and run
bash Miniconda3-latest-Linux-x86_64.sh
conda env create -f environment.yml # recreate env
conda env list # check if the new env installed properly
conda activate ENVNAME # activate the env
... # do your work
conda deactivate
```

For a list of useful commands, check the Conda cheatsheet.



- interactive web tool known as a computational notebook
- combines live code and multimedia resources in a single document
- rich interactive output incl. HTML, images, videos, LaTeX
- supports over 40 programming languages



- basically a Jupyter Notebook running in the Google cloud
- write and execute Python in your browser for free
- zero configuration, easy sharing, free access to GPU
- especially well suited to ML, data analysis and education

# Versioning

---



- by far the most used version control system
- open-source project created by Linus Torvalds
- fully distributed, supports offline work
- committing, branching, merging, etc. optimized for performance
- free hosting services are e.g. GitHub.com or GitLab.com
- my favorite guide: [rogerdudler.github.io/git-guide/](https://rogerdudler.github.io/git-guide/)

```
# installing git on Ubuntu
```

```
sudo apt install git-all
```

```
# usefull commands
```

```
git status # displaying the current status
```

```
git init # creating a new repository
```

```
git add . # staging latest changes
```

```
git commit -m "Commit message" # committing the changes
```

```
git push origin master # pushing the commit into origin
```

```
git pull # updating local repository
```

```
git clone # cloning some repository
```



## Weights & Biases

- framework agnostic tool for ML developers
- experiment tracking, dataset versioning, and model management
- hyper-parameter optimization, prediction visualization
- exploring and sharing of results from thousands of experiments
- in a centralized cloud dashboard

# Presentation

---



- turn jupyter notebook into a standalone web application
- have an interactive dashboard for your ML project in mintues
- simple installation with pip
- integrates with Jupyter Notebook or Jupyter Lab
- officially supported by the Jupyter Project

# Compute & Deploy

---



- environment to run your Jupyter notebooks from you git repo
- provides an easy way for your readers to run your code
- reviewers can reproduce your results without any overhead
- easy to set up and supports Docker
- create a button and place it in your readme.md



- fully-managed platform for simple app delivery
- not necessary to manage your own VPS
- supports most languages and deployment with Docker
- simple deployment with one git push command
- free plan (550-1000 hours of runtime per month)
- supports custom domain names on free plan



## Pavol Harar

pavol.harar.eu

Received an MSc in System Engineering and Informatics and a PhD in Machine Learning from Brno University of Technology. Gained experience in predictive modeling, signal processing, and parallel computing as a member of Brain Diseases Analysis Laboratory and Numerical Harmonic Analysis Group. Currently a postdoc at the Data Science Research Network @UniVie and a visiting postdoc at the Institute of Molecular Pathology in Vienna.