My little progresses in Computer Science, Machine Learning, Artificial Intelligence, etc...

Aritz Bercher

September 8, 2020

Abstract

This will be a diary containing a short summary of what I did every day of this fifth semester of master at ETH. I intend to learn programming and get some applicable skill in Machine Learning and Data Science. This will also help me gathering information found here and there.

Contents

1 Useful Resource

- Servesh showed me this blog where a guy posts summaries and explanation of advanced papers related to **new CS technologies**:
  `https://blog.acolyer.org/`
  It covers a broad range of topics. I might learn a log by simply trying to understand these posts.

- I found this page which gathers many good online resources for an **introduction to web development**:
  `https://www.fullstackpython.com/web-development.html`
- Rishu told me that one can use **severs for free** on this website:
  `https://www.heroku.com/`

2 Journal

05.10.17 $\sim$ **Starting with Linux, python, anaconda and jupyter notebook**
Creation of this diary. Two weeks ago I started studying a bit how linux works, in particular how to use the console. The slides and documentation provided by TheAlternative was very useful in order to make the first steps. Since a couple of days I try to assimilate the bases of Python. At first I was a bit puzzled by the fact that Python needs an interpreter and not a compiler (the difference is quite well explained in the book of Charles Severance p.8). I'm still a bit confused by the notions of Python virtual environment and in which way it differs from a virtual machine. I found a nice explanation there:
`https://blog.docker.com/2016/03/containers-are-not-vms/`
but it seems to be a reading a bit too advanced for my current knowledge. I started by installing Anaconda and it seems to be one of the easiest way to begin using Python.
I was a bit confused at the beginning before I had the following understanding of programming. First we write a script in a text file (generally inside a IDE) and then use a compiler (by pressing on a button in the IDE for instance) which translates it into some machine language and gives it to the CPU. And because of this I was failing to see how it could be done in several ways and what could be a virtual environment. Now I see it a bit differently. I see Python as a bunch of files (which include all the packages that we add little by little) describing a function which is called when we use `python arguments` in the terminal. So having a virtual environment is just telling to the computer "now when I say `python` in the terminal, I want you to use these files there".
It seems there is no standard editor for python script. I've been recommended to use jupyter notebook, pycharm and sublime. At first jupyter notebook didn't behave how I wanted. When I was in a virtual environment with python 2 installed and launched jupyter notebook from the command line, I was only able to create python 3 kernel. The problem was that I had not installed the package jupyter notebook in this virtual environment so it was going back to the root (the python installed "normally" on my linux) to find it. But then I installed the package jupyter notebook in all my virtual environment and it solved the problem. Today I learned what **encoding** and **unicode** is here:
`https://www.w3.org/International/questions/qa-what-is-encoding`
I learned a bit about **list**, **tuples** and **dictionaries** (globally data structures) here:
`https://docs.python.org/3.6/tutorial/datastructures.html#dictionaries`
and a bit about **class** and **methods** here:
`http://networkstatic.net/python-tutorial-classes-objects-methods-init-and-simple-example`

I also discovered the keyword `pass` which is explained here:
`https://stackoverflow.com/questions/13886168/how-to-use-the-pass-statement-in-python`
I also had difficulty to change the language of the dictionary of texmaker but I found the solution here (although it is supposed to be for windows):
`http://www.swisswuff.ch/wordpress/?p=166`

06.10.17 ~ yield, iterables, generators, with

I learned a bit about the keyword `yield`, **iterables** and **generators** here:
`https://stackoverflow.com/questions/231767/what-does-the-yield-keyword-do`
As I'm trying to create a python script to download songs from internet and that `youtube_dl` seems to be the right command to this in python, I tried to read this page:
`https://github.com/rg3/youtube-dl/blob/master/README.md#embedding-youtube-dl`
But for this I needed to understand what the keyword `with` was. I found several links:
`https://stackoverflow.com/questions/1369526/what-is-the-python-keyword-with-used-for#11783672`
`https://stackoverflow.com/questions/3012488/what-is-the-python-with-statement-designed-f`
`effbot.org/zone/python-with-statement.htm`
`https://docs.python.org/release/2.5.2/lib/typecontextmanager.html` `https://www.python.org/dev/peps/pep-0343/`
But these pages require already a good basic knowledge of python so I didn't get everything.
I learned here how to update firefox:
`https://askubuntu.com/questions/681312/how-to-update-firefox-on-ubuntu`

07.10.17 ~ vim, key binding with xbindkey, and linux permissions

I found a nice tutorial on vim which start's with basics:
Tutorial for vim
There is something which really annoys me. In linux I cannot use Ctrl+Alt+< to get the backslash. So I will try to follow the indications on this page (first answer):
`https://unix.stackexchange.com/questions/84707/how-can-i-make-ctrl-alt-act-like-alt-gr-i`
and this pages:
`https://wiki.archlinux.org/index.php/Xbindkeys`
`https://www.linux.com/news/start-programs-pro-xbindkeys`
While doing this I had to copy some files and I tried to do it in the terminal but I got the permission denied and I had to look again at this topic of permission. The toolkit v1 from slide 50 reminded me what "." and ".." was. Concerning permissions, this page is quite good:
`https://www.linux.com/learn/understanding-linux-file-permissions`
and this one is even better:
`https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Step_by_Step_Guide/s1-navigating-ownership.html`

Unfortunately it didn't work. I really tried to follow the first link up there
(`https://unix.stackexchange.com/questions/84707/how-can-i-make-ctrl-alt-act-like-alt-gr-`
)
but it didn't work. More precisely here is what I did

1) I installed what was needed with the command
   `sudo apt-get install xbindkeys xvkbd`

2) I copied one of the examples in the folder `/usr/share/doc/xbindkeys/examples` and into my `home` directory and called it `xbindkeysrc`. At this point I don't know if I should not put a dot before the name as seem to appear in the main tutorial.

3) I found the name associated to the combination of keys using the command
   `xbindkeys -k`

4) I edited this file and added the lines
   ```
   "xvkbd -xsendevent -text '\[backslash]'"
   m:0xc + c:94
   Control+Alt + less
   ```

5) I entered the command
   `xbindkeys -f  /.xbindkeysrc`

6) Seeing that it wasn't working I tried to execute directly the command
   `"xvkbd -xsendevent -text '\[backslash]'"`
   But this produces some warnings and error so I suspect that the command doesn't work.
   I think that someone posted a question on stack exchange for the same problem:
   `https://askubuntu.com/questions/158112/xvkbd-broken-with-warnings/211495`

08.10.17 ∼ Adding a password to jupyter notebook and various possibilities of bash

I would like to get rid of this page appearing each time I start a jupyter notebook where I have to insert the password. I found this page which might tell me how to do it:
`https://jupyter-notebook.readthedocs.io/en/stable/public_server.html`
but as I didn't find this jupyter folder, I wanted to use the command `find` but and as I would like to filter out all the "permission denied" errors, I read this page:
`https://unix.stackexchange.com/questions/42841/how-to-skip-permission-denied-errors-when-`
`42842`
I wasn't sure what was the symbol | for but I found it p.14 of the bashguide pdf given by TheAlternative.
I also found out what a symbolic link is there:
`https://kb.iu.edu/d/abbe`
As I didn't find the config file I created one as indicated and I received this message from the console indicating where the configuration file for jupyter was put:

```
Writing default config to: /home/aritz/.jupyter/jupyter_notebook_config.py
```

Even though I created this passord when I was inside the virtual environment my_python35 it seems that now when I enter the command `jupyter notebook` in this environment or on the normal one I'm directed to a page where I need to enter the password I chose (and no longer copy paste the code appearing in the terminal or clicking on the link appearing in the terminal).
Concerning youtube_dl I give up the idea of using it with python as it is completely buggy. I will instead look at the bash version which is in the bash exercise tutorial provided by TheAlternative.
The files bashguide and bash exercises provided by TheAlternative are very well done. I have the impression it gives most of the required knowledge in order to work alone. I learned the following things:

1. How to make executable scripts (using `chmod`) for instance.
2. I also understood that the tilde symbol $\sim$ (which appears often at the beginning of path names) designates the directory of the user (`/home/aritz/` in my case).
3. I learned about environment variables which allow the user to execute commands and somehow helps with the general configuration of the computer.
4. There is an exercise in the bash exercise file to create a directory where we can put all our scripts such that we can execute them from everywhere (or more precisely in every directory descendant of $\sim$).
5. Looking at these exercises I realized that a backslash in a bash script is just a line breaker.

09.10.17 $\sim$ First bash script and youtube-dl
I started reading the example in the bash exercise document of TheAlternative for youtube-dl. First I tried to understand what the `curl` command is and how it is used in the exercise. This lead me to learn a few things on the following topics related to how internet works:

1. protocols like http
2. URI and URL
3. POST request and web forms:
   `https://en.wikipedia.org/wiki/POST_(HTTP)`

I felt a bit overwhelmed by the man page of youtube-dl so I looked at this video:
`https://www.youtube.com/watch?v=1IFNHcs56ss`
and tried to reproduce what the guy was doing. But when I tried to do

`youtube-dl -F https://www.youtube.com/watch?v=pYtONwt_oNI`

I got an error. I looked online and I found this link:
`https://github.com/rg3/youtube-dl/issues/4232`
I uninstalled youtube-dl and reinstalled it using the following the "manual installation instructions" given here:
`https://github.com/rg3/youtube-dl/blob/master/README.md#how-do-i-update-youtube-dl`
but the first time it didn't work and I got the same error. I looked again online and all websites seemed to indicate that this error was appearing when one was using an outdated version of the utility. So I tried again to uninstall and reinstall and it worked. Maybe it is because I was in a conda virtual environment that I left (but I find it strange since I thought these virtual environments concerned only python). Or maybe it is because I used another window for bash... I don't really know. Strangely I don't find the man page for youtube-dl anymore. Maybe it was only in the outdated version. I found a page online:
`http://manpages.ubuntu.com/manpages/precise/man1/youtube-dl.1.html`
But I'm not sure if it is recent or not. Anyway one can always find help by typing `youtube-dl --help`.
I read that it's better to put .sh at the end of all bash script (easier to find then).
I created a script called `downloading_songs_youtube_bash.sh` which for now only downloads the audio associated the youtube video. I failed to implement options because I don't know how to access the url of the youtube webpage once I treated the video.
Edit: Actually it is explained in the bashguide how to parse options (p.13).
There is actually one script which does more or less the same as what I would like to do but for google chrome instead of firefox:
Downloading youtube videos from bookmarks with Google Chrome

11.10.17 ~ CIL and matrix factorization
I found this page for the course CIL given last spring which enumerates 3 projects:
CIL: Projects
But it looks like it's closed now.
I started reading the lecture notes of the CIL class. It has applications to Inpainting, Collaborative filtering, and Image compression. The main theoretical goal is to find ways to effectuate matrix factorization (approximation) efficiently, as many problems come down to this.
Roman sent me a link to a one of his projects on GitHub where he used python and youtube_dl:
Roman: video summerization youtube dl embeddings.

12.10.17 ~ Vectorized operations in Python: lists, Creating and manipulating matrices with NumPy, Comprehension lists, asterisk, zip, and other tools
I started looking at the exercises of the course CIL. Everything which follows comes

more or less from the first exercise sheet of CIL. I learned a few command to use NumPy objects and methods with the exercise sheet 1. The library NumPy seems to provide the basic tool to do array manipulation and vectorized operations which are computationally lighter for python that using for loops. If I understood well what is written here:
`https://stackoverflow.com/questions/11077023/what-are-the-differences-between-pandas-and`
`11077215`
NumPy provides the basis and pandas some more advanced tools.
Comprehension lists seems the tool to do a lot of vectorized operations, or to apply a function iteratively to entry of one or more lists. It is presented here:
`https://docs.python.org/2/tutorial/datastructures.html#list-comprehensions`
I learned about the zip function. Somehow, it inverses the dimensions in a list. `https://stackoverflow.com/questions/13704860/zip-lists-in-python#13704903`
The exercise 1 instruction sheet contains a list of basic vectorized operations and the associated commands in python.
I learned useful informations about the use of asterisk in order to take a list as input, and expand it into actual positional arguments in the function call, or deal with extra argument in a function, here:
`https://stackoverflow.com/questions/5239856/foggy-on-asterisk-in-python`
and there:
`https://stackoverflow.com/questions/400739/what-does-asterisk-mean-in-python`
The last exercise was nice. It consisted in computing for some points in $\mathbb{R}^2$ their likelihood depending on two different Gaussian model assumption and assigning each point to the most likely model.

13.10.17 ~ First glance at neural networks
I started looking at the slides of the course on Deep Learning.

14.10.17 ~ More arrays with NumPy, more on theoretical foundations of neural networks algorithms
I found this nice tutorial to use NumPy:
Tutorial for NumPy
and learned how to manipulate these arrays. I read the second lecture notes of Deep Learning, and learned a bit about Recurrent and MaxOut neural networks. It was quite theoretical, mostly approximation theory.

15.10.17 ~ Beginning of first ML project
I think I should decide a naming convention for my files and directories. I think I will always try to put Capital letters at the beginning of the name of a folder and lower case for files.
There is a nice explanation of what this SSH encryption is here:
Understandin the SSH encryption and connection process
I also started the first project of the course Machine Learning of ETH.

16.10.17 ~ ML project, trying to get started: Sumatra, Scikitlearn

I kept trying to do this ML project but there are plenty of things to get familiar with before being able to start like Sumatra and Scikitlearn (c.f. journal project ML1 for more details). I came across this page which gives a very good introduction to machine learning and how to use python for it:
basic tutorial on scikit learn

17.10.17 ~ Python coding guideline, Python modules and packages, python classes and metaclasses

I found this general guideline for python coding style:
`https://www.python.org/dev/peps/pep-0008/`
I learned a few things concerning python packages and modules in these two webpages:
importing subpackages
difference module and packages
For more informations about modules and packages this page looks very detailed:
`https://docs.python.org/2/tutorial/modules.html`
Concerning the Scikit-learn part of the project we are encouraged to read the following webpages:
coding guidlines
APIs of scikit-learn objects
rolling your own estimator
I think I start seeing why we use this scikit-learn. This story of fit, fit transform etc... is described in the "API's of scikit-learn objects" section. If I get it right, the organizers of the project want us all to use the same kind of syntax with the function/method/API that we use in our code.
I found another good website for python documentation: `http://www.pythonforbeginners.com/basics`
And in particular for docstrings:
docstrings
I will put all good tutorials for python in the latex file "how to use python".
This two webpages explain in a complementary way what a python signature is (more or less):
difference between function declaration and signature
how to read the signature of a function
There's a beginning of information concerning API's here:
`https://wiki.python.org/moin/API`
I found an interesting page about classes and magic methods here:
magic methods and operator overloading
I also found this great webpage about objects, classes, and metaclasses in python:
`https://stackoverflow.com/questions/100003/what-is-a-metaclass-in-python#6581949`

18.10.17 ~ More on scikit-learn
The organizers published a very helpful sequence of slide which I called

`final_submission_instructions`

which explains a bit better how to use this scikit-learn framework.

19.10.17 ~ Trying to follow the instructions provided for the ML project, git, pipeline, quit frozen sublime, gitlab
The command to launch sublime text from the console is `subl`, but if sublime text gets frozen (which happened today) the command to close it is:

```
sudo killall sublime_text
```

I mangaged to add a shortcut to sublime text in order to be able to comment and uncomment several lines at the same time with ctrl+T following this guideline:
stackoverflow
Here is a nice tutorial about git and plenty of more in depth references at the end:
`http://rogerdudler.github.io/git-guide/` Otherwise I finished following the instructions given by in the file called "final submission instructions". I really didn't do much as everything was implemented but I got a bit an idea of what these different tools are. There are still plenty of things I need to figure out (see my ml project journal for more details).
Concerning scikit-learn pipeline, I found these two pages which seem very good:
`https://stackoverflow.com/questions/33091376/python-what-is-exactly-sklearn-pipeline-pip`
`33094099` and `http://scikit-learn.org/stable/auto_examples/model_selection/grid_search_text_feature_extraction.html`
There is also a presentation of gitlab here:
Introduction to gitlab

24.10.17 ~ Flake8, Submitting my code for ML project 1
I will try to submit my code (even though it is exactly what the tutors gave us as a model). I ran the command `flake8` and corrected what was pointed out. It seems to indicate style errors (like additional spaces at the end of a line) in order to obtain a code which follows some standards. Here is a webpage which introduces it well:
Flake8

26.10.17 ~ Advanced Computational Statistics webpage, Scikit-Learn tutorial, Support Vector Machine algo
I've been catching up the lectures of Deep Learning but it's fairly theoretical. This morning I had the course of Meinshausen about hidden Markov chains and he showed us a nice simulation of a drone which would have to guess its position in a maze. I should try to reproduce the algo in python. He said he would put his R script on the website (which can be found here:
`http://stat.ethz.ch/lectures/as17/adv-comp-stats.php`).

I started to read the first scikit-learn tutorial:
`http://scikit-learn.org/stable/tutorial/basic/tutorial.html` and I stumbled upon the "Support Vector Machine" algorithm described here:
support vector machine which is strongly related to the kernel trick explained by Buhmann (which can be explained here: Kernel method)
I also learned about perceptron (algorithm) already met in the ETH Deep Learning class:
`https://en.wikipedia.org/wiki/Perceptron`,
and the kernel perceptron method:
`https://en.wikipedia.org/wiki/Perceptron`
This last article gives a good introduction and intuition of the Kernel method.
But the choice of the kernel is probably very difficult and requires certainly some very good insight.

27.10.17 ~ Stamtish in linux, and parsing option in bash script
Today I went to a "Stamtish" of "The alternative" to try to fix two issues:

1. using Ctrl+alt+less to obtain a backslahs
2. the pointer keeps jumping when I type because I touch the edge of the touchepad involuntarily.

To solve the first problem the closest thing we found was the command

```
xmodmap -e 'keycode 108=Alt_L Control_L'
```

but it doesn't really work since then altgr doesn't anything anymore. And it's temporary meaning it disappears each time I log out. To make it permanent, Leonid Block told me I should include the line in my profile file (somewhere in the home directory I guess).
For the second problem, I could maybe change the position of the top right edge of my touchpad looking at thispage:
`https://www.x.org/archive/X11R7.5/doc/man/man4/synaptics.4.html`
Beside, I went through the documentation of the alternative and I saw that in the bashguide that there is a way to actually a tutorial to parse option (p.13).
I looked again at the tutorial with xbindkey mentionned above but it looks like the command

```
xvkbd -xsendevent -text '\[backslash]'
```

itself produces errors. From what I read online, it seems to be a general problem. I should post a question on stack-exchange

28.10.17 ~ Making your own templates for anki cards
With the help of the anki online manual:

Anki manual
I found how to make my type of cards, and I created a new type of cards called "Type answer" where I have to type the answer to the question. This will be handy to memorize programming commands. The steps are

1) In the Deck menu click on the button "Add".

2) In the window appearing click on the button corresponding to the type.

3) In the window appearing click on the button "Manage".

4) In the window appearing click on the button "Add".

5) In the window appearing select "Clone:Basic".

6) In the window appearing type the name of your new deck.

7) Go back to the Deck menu and select Browse. In the column to the left, select the newly created card type template.

8) Select a card and press the "Cards..." button and edit the template.

30.10.17 ∼ Trying to use typebox in Anki, creating templates for Anki cards, a bit more of Scikit-learn
I tried to create templates with Anki where one can type an answer on several lines but a priori it's not possible. I found an add-on online (here) which is supposed to solve this issue but it doesn't work. I will post a question on this forum:
`https://anki.tenderapp.com/welcome`
as soon as I will have received the email to confirm my new account.
With Latex, I received an error

```
! Package babel Error: You haven't defined the language french yet.
```

which was most likely due to the babel package for a file that I had duplicated from windows which worked on windows. Deleting the .aux file solved the issue.
I kept reading the scikit-learn tutorial. It refreshed my knowledge about logistic regression. I also learned that:
"For many estimators, including the SVMs, having datasets with unit standard deviation for each feature is important to get good prediction."
It also seems that these kernel estimators are very important.

02.11.17 ∼ Cross-validation and renormalization
I spent the last day studying the tutorial of scikit-learn and in particular how to do cross-validation with python. It seems I obtained very different results for the exercise with diabetes, depending if I shuffle the observations in the data set or not. Next thing I should look at is the normalization of variables. I found this nice topic on Quora: Quora: normalization in machine learning. It seems to be very important.

03.11.17 ~ Difference between Machine Learning and Probabilistic Artificial Intelligence

I think I understood the fundamental difference between the topics covered in ETH Machine Learning and in ETH Probabilistic Artificial Intelligence. In both cases we are interested in making inference and estimating probabilities or distribution. But in the first case, we have very little idea of how the observed features are related to the response variable. For instance the relation between the values of the voxels (pixels in 3D) of an MRI image and the state of the brain. In the second case we have a clear idea of the causal relations between the different variables (observed or hidden) that we have (for instance the example of the alarm-burglary-earthquake-neighbors system), and we can use this knowledge about this structure (often represented by a graph) in order to do some inference or estimation. I also think that in the first case, we assume to have a lot of data, whereas in the second, I guess we need less (maybe even not at all). Also as Sid (Sidarta from ML) pointed out, these kind of Bayesian networks might be useful for designing some robots or systems. But for raw data analysis, it's what we see in ML which seems more appropriate.

I started looking at an example in the scikit-learn documentation on how to use pipeline but I'm a bit confused by the variables h and w. Maybe it is height and width. Each observation is a matrix instead of a vector. After looking at the set, the data is given in two forms: either in `lfw_people.data` where each observation is a vector of lentght 1850, or in `lfw_people.image` where each observation is 37×50 matrix.

04.11.17 ~ Principal Component Analysis (PCA) and Singular Value Decomposition

PCA comes again and again, so I think I should try to learn it properly. I quickly reviewed what I had already seen in the first lecture of CIL. This being said I'm not use that it is exactly the PCA, since it's called SVD (Singular Value Decomposition). I read this very interesting page which gives a good explanation of the two different quantities and their relations:

Stack Exchange: PCA and SVD

I should later I'll look at this page of sk-learn:

sk-learn: PCA

06.11.17 ~ Beginning of second project of ML course

I started yesterday the second project in machine learning. I found this nice scikit learn cheat sheet which explains which algorithm use when:

sk-learn cheat sheet

16.11.17 ~ ML second project and interesting links for sklearn classification

I've been working on the second project of the course Machine Learning which is about classification, with the special feature that the response variables are probabilities to belong to one of four categories instead of being the actual category (c.f. the journal for

this project for details).
I found this link which could be useful and is part of a global course on machine learning I think:
Classification with scikit learn

18.11.17 ~ Features selection and preprocessing
I found this explanation for git commit and git push:
Stack Overflow: difference git commit git push
I found this blog for preprocessing:
blog about feature selection and preprocessing

19.11.17 ~ Nikos recommendation for deep learning
Nikos recommended me two links for deep learning:
BiDAF
GitHub: e2e-coref

22.11.17 ~ Code Submission in my ML project and Git GitHub Tutorial
I submitted the code for the second project of machine learning and then I started reading this (so far very good) tutorial:
Git and GitHub for beginners

23.11.17 ~ More on Git and GitHub
I'm trying to follow the instructions on the link cited above. But there is a first issue: I have a local username and password for git inside the repository of my second project of ML, as we can see in this copy paste of my shell:

```
aritz@geronimoT460p:~(...)/ml-project$ git config --list
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=https://gitlab.vis.ethz.ch/vwegmayr/ml-project.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
user.email=abercher@student.ethz.ch
user.name=11-810-736
remote.submission.url=git@gitlab.vis.ethz.ch:vwegmayr/ml-project.git
remote.submission.fetch=+refs/heads/*:refs/remotes/submission/*
```

Then from what I understood there is a mechanism of encryption for the files shared between my computer and the servers of GitHub or GitLab which relies on an "ssh key". Concerning ssh keys, I read this:

git and SSH
I added my ssh key to GitHub following these instructions:
Adding a new ssh key to your github account
If I understood right what's written on this page:
Stack Overflow: Where dod the settings in my git configuration come from?
I think that the global settings are overwritten by the local ones. The result is strange. When I look at the config inside the repository of the project, I have two user.name and user.email:

```
aritz@geronimoT460p:~$ git config --list
user.name=abercher
user.email=abercher@outlook.com
aritz@geronimoT460p:~$ cd Documents/(...)/ml-project/
aritz@geronimoT460p:~/(...)/ml-project$ git config --list
user.name=abercher
user.email=abercher@outlook.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=https://gitlab.vis.ethz.ch/vwegmayr/ml-project.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
user.email=abercher@student.ethz.ch
user.name=11-810-736
remote.submission.url=git@gitlab.vis.ethz.ch:vwegmayr/ml-project.git
remote.submission.fetch=+refs/heads/*:refs/remotes/submission/*
```

But it seems that the local ones prevail:

```
aritz@geronimoT460p:~/(...)/ml-project$ git config user.name
11-810-736
aritz@geronimoT460p:~/(...)/ml-project$ git config user.email
abercher@student.ethz.ch
```

I started reading this page (second page of the previous tutorial):
GitHub for beginners part 2
The part in the tutorial where I do my first git push didn't work. I tried to follow the instructions given in the error message:

```
git config --global push.default matching
```

but it produced again an error when I tried (again) to do

```
git push
```

so instead I followed some instructions found on this page:
Stack Overflow: no refs in common and non specified doing nothing
i.e. I entered

```
git push origin master
```

and it worked.

25.11.17 ~ Quora list machine learning algorithms
I found this question and answer on Quora, about what are the important machine learning algorithms:
Quora: list ML algo
It seems to be quite complete and could be use in order to find some ideas of algo to use.

26.11.17 ~ Performance of different algorithms
With this second project of ML, I have the impression that in some cases, different algorithms and approaches can yield very similar results. In a way, it looks like there is an amount of true information which can be learned and then it's probably more about how we preprocess the data and make good use of cross-validation to tune the parameters than about picking one approach or the other.

27.11.17 ~ XGBoost
Some student posted on plazza a link for multi-label classification:
multi-label-classification
The blog seem to contain plenty of information for ML.
I also learned about XGBoost which seems to be some libraries to use trees intensively:
Quora: What is XGBoost

04.12.17 ~ Reservoir algorithm
I wanted to use an algorithm which samples k lines at random from an array with more than k lines (in order to reduce the unbalance of my data set of the project of ML 3). So I looked online and this is called Reservoir algorithm. It turns out to be related to big data. I found several interesting links. This one explains the idea quite well:
Stack overflow: design a storage algorithm
And this one gives a simple implementation:
Stack overflow: reservoir sampling

17.12.17 ~ Detecting seasonality in time series
All these previous days I've been working on the third project of machine learning where we have to classify these electro cardiograms (ECG) signals. Discussing with Nicola, I realized that I should have looked at tools of time series earlier like autocorrelation. I

stumbled upon a nice explanation of how to detect seasonality inside a time series here: stack exchange: detect seasonality

31.12.17 ~ End of ML project on ECG, Beginning of fast AI course on Deep Learning

The semester ended a bit more than a week ago, and with it the last project of ML on ECG. My best idea was to reconstruct the signals on its zero part by duplicating the non-zero part and using (local) auto-correlation to find the length/duration of one heart-beat and the regularity of the signal. What I did wrong was to give up too early on finding some existing tools for the task. I didn't include key-words like python library in my google searches.
I started yesterday an online course called fast AI recommended by Roman, about Deep Learning:
`http://www.fast.ai/`.

02.01.17 ~ KDE, KDE Plasma, Kubuntu, Copy-paste only with selection, Wine

As I was looking for some informations concerning keybard short-cuts to open a terminal in case my graphical interface crashes, I tried to understand a bit more about KDE, Kubuntu, etc... In fact KDE is the name of a community developing open and free soft-ware (Wiki: KDE), kubuntu "is an official flavour of the Ubuntu operating system which uses the KDE Plasma Desktop instead of the Unity graphical environment" (as explained here: Wiki: kubuntu).
I also learned that one can do copy paste by selecting the part we want to copy and then using the middle button to paste.
I found the forum kubuntuforums.net which seems to contain plenty of nice informations.
I learned about the existence of Wine "a free and open-source compatibility layer that aims to allow computer programs (application software and computer games) developed for Microsoft Windows to run on Unix-like operating systems".
I asked these two questions on kubuntuforums.net:
`https://www.kubuntuforums.net/showthread.php/72832-Can-t-find-help-button-in-Dolphin?p=408556#post408556`
`https://www.kubuntuforums.net/showthread.php/72833-Launching-a-terminal-if-GUI-crashes?p=408560#post408560`

03.01.17 ~ Use console in linux when GUI crashes

Some one told me on kubuntuforums.net that one can use Ctrl+Alt+F6 to get a text console at any time. Then I have to log in with my username aritz. And to get out it is Ctrl+Alt+F7. Another person told me how to enable the short-cut to have be able to start a terminal with Ctrl+Alt+t

I installed Skype and used my outlook account to log in, but it doesn't seem to be the same as my previous account.

15.01.18 ~ Usefulness of loss function
At the end of the notebook of the first lesson of the course on deep learning (part 1 v1) of fast.ai, they explain why one should look at a loss function and not only the accuracy. Basically, the accuracy of a method doesn't take in account how certain the algorithm was about its predictions. A loss function may encapsulate this better.

20.01.18 ~ Coming back to downloading songs from youtube
I came back to my first project which was to download the songs of the youtube page that I bookmarked. I looked again at this page:
github: embedding youtube-dl
and it looks like what I though was a command for python is a command line for the bash, that we can embed in a python script.
I downloaded plenty of songs using the command:

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one was "The first station - Gangsta".

23.01.18 ~ Downloading more songs from youtube, Recommendations of Roman
I downloaded plenty of songs using the command:

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one was "Debussy - Rêverie - YouTube".
Roman recommended several resources to me.

- A news aggregator for machine learning: `https://news.ycombinator.com/`
- A course quite theoretical: `https://sites.google.com/view/deep-rl-bootcamp/lectures`

26.01.18 ~ Downloading more songs from youtube
I downloaded a few more songs from my bookmarks with the command

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one was "Clint Eastwood" from Gorillaz.

08.02.18 ~ Trying to extend my kubuntu partion, creating a live key, reinstalling everything
As I announced it in my linux journal, I wanted to increase the size of my linux partition but it didn't go as smoothly as I expected and it took me a week and a half. When I tried to move my kubuntu partition a problem occured and my whole kubuntu partition was corrupted. I had to delete it and recreate it. The main steps (that I remember were):

1. Create the usb live key. There was a command with `dd` for this. But one has to be SUPER CAREFUL and use the right disc name otherwise one can wipe out the content of another partition.
2. Connect my windows activation to my outlook email account to be able to always redownload Windows if necessary. I also did an .iso file for linux that I put on my hard drive.
3. Whithin Windows shrink as much as possible the Windows session. Doing it via KDE partition manager might be dangerous for Windows.
4. After changing the priorities in the bios to have my live usb key first I could start a "testing session of kubuntu". Whithin it, using KDE partition manager, I could delete the problematic partition containing kubuntu, delete linux swap. After that I created 4 new partitions:
   (a) 25GiB for root partition, format ext4
   (b) 160GiB for the home partion, format ext4
   (c) 4GiB for the swap partion, format linuxswap
   (d) 20GiB for a backup partition, format ext4
5. Then I followed the instructions of the installer on the live key.
6. I had to give the priority to the UEFI mode in the bios (the priority was given to the Legacy mode previously).
7. Update grub with

   ```
   sudo update-grub
   ```

This wouldn't have been possible without the help of the poeple on kubuntuforums.net and this specific tread:
kubuntuforums: Decreasing windows partion's size and increasing kubuntu partion's size
mr_raider also gave me this link concerning the switch between graphical card:
`http://ubuntuhandbook.org/index.php/2016/04/switch-intel-nvidia-graphics-ubuntu-16-04/`
I followed the instructions but when I use the NVidia one, I have some glitches. There is a little square of text which keeps appearing next to the mouse pointer. For now I use the intel one.
I also had to reinstall everything and change the settings. The fisrt thing I had to do was to make everything look bigger. For this there are two things to change in the setting: font ¿ font (make it 192), and display and monitor ¿ display configuration (scale display).
Following the advice of Qqmike I entered

```
sudo efibootmgr -v
```

in the terminal and saved the output inside a file I called efibootmgr.

Concerning the BIOS and the UEFI, I learned that the BIOS is an old software and that UEFI is the new "firmware" which is more recent and generally better. But it seems that if UEFI isn't able to boot because it doesn't find "EFI Service Partition to boot from" then it goes back to BIOS mode. There is a tread on quora about this:
Quora: difference UEFI and Legacy Mode

12.02.18 ~ Fixing the problem with little rectangle appearing near the pointer of the mouse
There was an annoying little rectangle which kept appearing near the pointer of my mouse, but I found the solution here:
`https://bugs.launchpad.net/ubuntu/+source/nvidia-graphics-drivers-384/+bug/1684240`
which is:
"My workaroud for this Problem: Go to KDE System Settings -¿ Display and Monitor -¿ Compositor (left side) -¿ set "Rendering Backend" to XRender"
and it worked for me as well.

13.02.18 ~ Getting more familiar with Kaggle, Titanic example
I looked a bit at the titanic example of Kaggle here:
Kaggle: titanic
I learned a bit about **kernels**.

14.02.18 ~ Changing texmaker dictionary, extracting some pages of a .pdf file
Since I reinstalled kubuntu, I don't have french dictionary for texmaker anymore. So I went into my Windows partition and sent myself an email with the dictionaries. Then I had to make them writable and readable with

```
sudo chmod o+w fr_FR.dic
```

The French dictionary works but then it turned out that the English dictionaries are not working anymore.
I followed this tutorial:
`http://www.swisswuff.ch/wordpress/?p=166`
but it didn't work. If the "hyph" type dictionary, there is no correction at all, and with en_us.dic, everything is underline in read.
Then I followed the instructions on this page:
`https://copiancestral.wordpress.com/2012/02/13/texmaker-error-cant-open-the-dictionary/`
and moved both the .dic and .aff to the folder

```
/usr/share/myspell/dicts
```

(like I had done before) and made both the .dic and .aff readable writable and executable with `chmod` (as above) and after changing the dictionary, it works!

I used `pdftk` in the terminal to extract a specific page of a .pdf file and make a file out of it with the command:

```
pdftk Brochure_for_students_New.pdf cat 13 output Brochure_dfine_page_New.pdf
```

15.02.18 ~ Reinstalling sublime text
I reinstalled sublime text following this guideline:
`https://www.sublimetext.com/docs/3/linux_repositories.html`

19.02.18 ~ One-hot-encoding vs label encoding
I learned a bit about one hot vector **representation vs label encoding representation** here: stack exchange: one hot vector representation vs label encoding representation

20.02.18 ~ Reinstalling Cisco Anyconnect/ETH VPN
I reinstalled the Cisco Anyconnect tool following the indications given there:
`https://www.ethz.ch/content/dam/ethz/associates/services/Service/IT-Services/files/service-desk/guides/vpn-en.pdf`

25.02.18 ~ SQL servers, Relational database, Management system, problems with youtube-dl
I learned a bit about SQL servers here:
`https://www.databasejournal.com/features/mssql/article.php/3769211/What-is-SQL-Server.htm`
I tried to download songs from youtube with

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

but I ran into trouble. I uninstalled youtube-dl and reinstalled it following the instructions on the github page of the program. But it failed. I posted a question on kubuntuforums:
kubuntuforums: youtube-dl not found

26.02.18 ~ Solving the problem with youtube-dl
With the help of the guys of kubuntuforums I discovered that I had youtube-dl in an old conda environment (my_python35). After deleting I ran

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

and it worked! I downloaded a few more songs. The last one was Eminem - Business.

05.03.18 ~ Downloading more songs from youtube
I downloaded more songs from youtube with the command

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one I downloaded was "M83 'Midnight City' Official video - YouTube"

12.03.18 ~ Solving the French babel issue for latex
I had a problem with scripts which were previously working on windows. Somehow the option

```
\usepackage[french]{babel}
```

but I found the solution to my problem here:
stackexchange: package babel error unknown option francais
Then I just had to install a package with the command line:

```
sudo apt install texlive-lang-french
```

and it worked again.

12.03.18 ~ Downloading more songs from youtube
I downloaded more songs from youtube with the command

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one I downloaded was "Tom Jones - I'll Never Fall In Love Again"

14.03.18 ~ Trying to download pycharm, Recurrent Neural Networks, Starting a journal for the theoretical aspects of machine learning
I went on the website from pycharm and applied for a student license. But after I agreed to the term of agreements and made a JetBrain account, I still arrive on this page:
`https://www.jetbrains.com/pycharm/download/#section=linux`
I don't know which version to choose. I will ask Nikos.

I read this introduction to **Recurrent Neural Networks in NLP**, and their applications to **Language Modeling and Generating Test**, **Machine Translation**, **Speech Recognition**, **Generating Image Descriptions** (when coupled with some CNN):
`http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-`
I also read this page on Long Short Term Memory (LSTM) networks:
`https://colah.github.io/posts/2015-08-Understanding-LSTMs/`
which is a very successful special case of RNN.
I created a journal for the **theoretical aspects of machine learning**.

16.03.18 ~ Connecting to eth network
To connect to the "eth" wifi, I downloaded a small document (called "wlan-eth.pdf") which contains the required information. The username I had to use was abercher@student.ethz.ch and the password is the same as for connecting to the VPN.

20.03.18 ~ Starting a journal for Pytorch
I created a new journal, which is dedicated to **Pytorch**.

23.03.18 ~ Installing pycharm
I installed pycharm by first entering in the terminal

```
tar -xvzf pycharm-professional-2017.3.4.tar.gz
```

and then executed `pycharm.sh` with

```
./pycharm.sh
```

Then I had to choose a few option (I didn't add the VIM plugin) and a window opened where I can choose between

- Create new project
- Open
- Check out from version control

03.04.18 ~ Downloading more songs from youtube, Starting a jounral for TensorFlow
I downloaded a couple of new songs from youtube, the last one being "In my Bed" of Kid Francescoli, with

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

I created a new journal for **TensorFlow**.

14.04.18 ~ Downloading more songs from youtube
I downloaded a couple more songs from youtube. The last one is "Asleep" from the Smiths. I used

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

29.04.18 ~ Downloading more songs from youtube
I downloaded a couple more songs from youtube. The last one is "One more light" from the Linkin Park. I used

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

01.05.18 ∼ **Downloading more songs from youtube**
I downloaded a couple more songs from youtube. The last one is "Run boy run" from the Woodkid. I used

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

02.05.18 ∼ **Starting a journal for chat bots**
I created a new journal, which is dedicated to **chat bots**.

03.05.18 ∼ **Log-space**
Looking at this Pytorch tutorial:
`https://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html`
I stumbled upon the definition of **Log-space reduction**:
Wiki: Log-space reduction
but didn't understand it very well because my knowledge in complexity theory is quite small.

04.05.18 ∼ **Sentiment analysis with Pytorch**
I started a project where I will try to do sentiment analysis with Pytorch for IMDB reviews (like in the notebook of fastai).

06.05.18 ∼ **Downloading more songs from youtube**
I downloaded a couple more songs from youtube. The last one was "I apologize" by Bobby Vinton. I used

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

15.05.18 ∼ **NLP coursera course**
I started a **coursera course on NLP**:
`https://www.coursera.org/learn/language-processing/lecture/XZtoZ/hashing-trick-in-spam-f`
and a journal for it.

16.05.18 ∼ **Hash functions**
I learned a bit about **hashing function** on wikipedia:
`https://en.wikipedia.org/wiki/Hash_function`

17.05.18 ∼ **Vi vs Vim, Vi and Sublime text**
Nikos told me that he was using Vim on the top of Pycharm. I was wondering if I could

do the same with Sublime text. I discovered that **Vim is not the same as Vi** here:
`https://askubuntu.com/questions/418396/what-is-the-difference-between-vi-and-vim#418413`
and that one can use Vi inside sublime text, as explained here:
`https://www.sublimetext.com/docs/2/vintage.html`

19.05.18 ~ Downloading more songs from youtube
I downloaded a couple more songs from youtube. The last one was "Note blanche" by N'to. I used

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

22.05.18 ~ Downloading more songs from youtube
I downloaded a couple more songs from youtube. The last one was "My love" by Kovac. I used

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

13.06.18 ~ Doing back-ups, using rsync, transforming a Linux USB live key into a normal USB key
I read this tutorial:
`https://www.howtogeek.com/135533/how-to-use-rsync-to-backup-your-data-on-linux/`
The beginning explains how to do backups on a hard drive and it seems quite easy. I tested it and it works.

I transformed the Linux live key I had made to reinstall linux back into a normal USB key that I can use for file transfer following the indications of the users of Kubuntuforum:
`https://www.kubuntuforums.net/showthread.php/73778-Erasing-a-Linux-live-key?p=416481#post416481`

28.06.18 ~ Downloading more songs
I downloaded more songs from youtube using

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one was Creep from Radiohead.

27.07.18 ~ Downloading more songs from youtube
I downloaded more songs from youtube using

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one was Vision One from Röyksopp

08.08.18 ~ Unit tests, Websites to prepare coding interviews
I read a bit about "**unit tests**" here:
Stack Overflow: What is unit testing and how do you do it?

Nikos also gave me these three website as references to **practice coding for interviews**:

- `https://leetcode.com`
- `https://www.geeksforgeeks.org/`
- `https://www.hackerrank.com/`

Actually, one has to pay to see the solutions. I will take a subscription for a month for leetcode.com but I need to remember to unsubscribe:
`https://leetcode.com/subscription/`
it even looks like if I unsubscribe now, I can still use it for a month.

16.08.18 ~ New projects, ideas of different things I should do
I need to prepare for the next coding interviews I will have. It seems that there are different things I can do:

1. In order to prepare for the challenge of Oto.ai, it would be good to make a little project where I would have to tackle a task that I don't know very well and deal with new data. For this I thought that a little project with audio signal could be good. I could put myself in real conditions and give myself a day to do the best I can. Roman gave me the idea of trying to remove noise from audio signals recording voices.
2. I could try to find websites giving realistic challenges (not just algorithmic ones).
3. I could try to learn Vim as it could help me being faster.
4. I could try to look at this co-occurence thing as it seems to be a good unsupervised technique and I know nothing about it.

I also downloaded songs (using my new bash script data_downloader.sh). The last one was "Here she comes again" of Röyksopp.

14.07.18 ~ Creating new Anki template for coding
Using the advice given on this page together with what I had saved in this journal on

the 28.10.17, I created a new template for my Anki cards, which is aligned on the left. This is much more readable as it preserves the indentation (more or less).

05.09.18 ~ Downloading Swing songs
I downloaded some swing Songs with

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

the last one being from Louis Prima, Swing, Swing, Swing

08.09.18 ~ Downloading more songs
I downloaded some new songs

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

the last one being Harlem Shuffle

20.09.18 ~ Downloading more songs
I downloaded some new songs

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

the last one being Booty Swing of Parov Stellar

02.11.18 ~ Graph data base with neo4j
I discovered the **neo4j** and its query language **Cypher** which are presented for instance on this page:
`https://neo4j.com/developer/cypher-query-language/`
It allows to build some nice graphs. It looks like there is a free version for young start-ups.

04.11.18 ~ Downloading more songs
I downloaded more songs with

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

the last one was Randy from Justice.

28.11.18 ~ Updating youtube-dl
I updated youtube-dl (I had errors) following the indications on this page:
`https://github.com/rg3/youtube-dl/blob/master/README.md#how-do-i-update-youtube-dl`
and it worked.

02.11.18 ~ New iPod and new problems
Last time I put the songs on my iPod I went running afterwards but then when I came back I put my iPod in the pocket of my pants and put the pants in the washing machine...

I bought a new one on Ricardo, but I had a lot of trouble to put music on it. First, I had an error message when I was trying to add songs from rhythmbox to it (something with read only). Then I went in windows, installed iTunes, "restored" the iPod and tried again. This time the error message had disappeared but once I was unplugging it from my computer, no songs seemed to be on the iPod. Eventually I managed to put my songs on the iPod by:

1. Put all the songs in a same folder
2. Drag the folder to the library of iTunes.
3. Use the automatic refill from iTunes.

But it would be much nice to do it from Rhythmbox.
I created a thread on Kubuntu forum:
`https://www.kubuntuforums.net/showthread.php/74794-Can-t-transfer-songs-from-Rhythmbox-t`
`p=423375#post423375`

02.01.19 ∼ **Downloading more songs**
I downloaded more songs with

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

the last one was L'hymne à la joie de Beethoven.

04.01.19 ∼ **Regex**
I found this page very useful to understand **regular expressions** and learn to use them:
`https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f`
Edit (07.11.19): I found that these pages give a clear and concise overview of what can be done:
`https://www.rexegg.com/regex-quickstart.html`
`https://www.rexegg.com/regex-disambiguation.html#lookahead`
`https://www.rexegg.com/regex-lookarounds.html#password`

14.01.19 ∼ **Encoding**
I read this page about what is **encoding**:
`https://www.joelonsoftware.com/2003/10/08/the-absolute-minimum-every-software-developer-`

09.03.19 ∼ **Downloading more songs**
I downloaded more songs with the command

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one was "The day we fell in love" from The ovation.

27.05.19 ~ Downloading more songs

I downloaded more songs. The last one was Go Down North of Serafyn.

15.06.19 ~ Downloading more songs

I downloaded more songs using

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one from the Music folder was **1492 Conquest of Paradise** of Vangelis. The last one from the Lindy Hop folder was **Mack the Knife** of Louis Armstrong.

25.06.19 ~ How does Internet work

I found this very nice page:
`https://www.lifewire.com/web-browsers-and-web-servers-communicate-817764`
which explains (together with the many other pages of this website) concepts like

- **How a web browser communicates with servers**
- **URL** (Uniform Resource Locator), which are of the form

  ```
  protocol :// host / location
  ```

- **IP addresses**, which are of the form

  ```
  151.101.65.121
  ```

- **HTTP** (and HTTPS) as well as other protocols like **TCP/IP**
- What is a **router**
- **DNS** (Domain Name System) which "translates internet domain and host names to IP addresses and vice versa"

This website gives a good explanation and illustration of what is an **HTTP request** (without focusing on a specific type like GET, PUT,...):
`https://www.toolsqa.com/client-server/http-request/`
it also provides a good illustration of **HTTP response**:
`https://www.toolsqa.com/client-server/http-response/`

Edit (15.11.19): See the entry of the 15.11.19 for more resources (that I haven't read yet).

Edit (16.11.19): I read this very good post giving an overview of the technologies used when one person sends an email to another (the post is called **How internet works**):
`https://thesquareplanet.com/blog/how-the-internet-works/`
It presents the following things:

- internet server
- Internet Protocol (IP) Address
- Domain Name Service (DNS)
- Dynamic Host Configuration Protocol (DHCP)
- Gateway
- Packets
- Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)
- Internet Message Format (IMF)
- Text Encoding
- Simple Mail Transfer Protocol (SMTP)
- Hypertext Transfer Protocol (HTTP)
- Application Multiplexing
- GET and POST requests
- HTML document
- Javascript
- Cookies
- https and Transport Layer Security (TLS)
- Different internet layers or tiers (1, 2, 3)

Edit (30.01.20): I read this post on stack overflow explaining what **port forwarding** is:
`https://superuser.com/questions/284051/what-is-port-forwarding-and-what-is-it-used-for#284073`
Edit (30.01.20): Sarvesh explained me what is an **application server** and what is the difference with a **web server**. What I understood is that at the beginning servers of the world wide web would only provide static content (premade files) generally an .html file, or .js file (or similar format). It's ok if a page displays the weather, but if a

client asks for a piece of data recorded in a data base, it would be too complicated to have to display every possible entry in a static way. A framework allowing this kind of process is called a **web server** framework. If one wants that a software engine processes the answer and returns something, one uses an **application server**. **Gunicorn** is such a framework which is specialized in running python code. According to Sarvesh, its strength compared to **flask** (which is also an application server framework), is that it scales much better.

Edit (23.04.20): I learned a bit about **webhooks**. From what I understood by looking at a few tutorial, it is a tool which allow to automatically send some data somewhere when some kind of event happens in a web application. The data sent describes the event inside a formated jason. I guess that it can be used in plenty of different contexts and that the details vary, but the recurring example is the notifications one would receive on his mobile phone if some payment was made on his bank account.
I also learned about **web templates** or more commonly **templates** which are pretty well explained on wikipedia:
`https://en.wikipedia.org/wiki/Web_template_system`
I also learned about **static web pages** or more commonly **static pages**. Wikipedia explain them well:
`https://en.wikipedia.org/wiki/Static_web_page`
Edit (24.04.20): I learned a bit about **url slugs** here:
`https://en.wikipedia.org/wiki/Clean_URL#Slug`

14.07.19 ~ Downloading more songs
I downloaded more songs with the command

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one from the Music folder was **Video killed the radio stars** of the Buggles.

25.08.19 ~ Tutorial about spell checking

I just finished reading this very good tutorial about **spell checking** and how to implement an spell checker. There is a toy model which presents some of the key ideas:
`https://norvig.com/spell-correct.html`
It rests on a Bayesian approach. Conclusions are really interesting. One the conclusion is that if one wants something really efficient, one should have a language model inspecting the whole sentence and not only the words as presented here. Another one is that one should implement it in a compiled language instead of in an interpreted one (what's the difference?). I guess something similar could be used to correct the ASR output.

08.09.19 ~ Downloading more songs

I downloaded some new songs with

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

The last one was "If you seek Amy" of Britney.

10.09.19 ~ Deploying effectively ML models in industry

I read this article:
`https://towardsdatascience.com/solving-machine-learnings-last-mile-problem-for-operation`
At the end it mentions some **Decision Management** or **Digital Decisioning Platform** softwares which can be helpful. But unfortunately, it doesn't mention any specific one.

02.10.19 ~ API vs SDK

I read this page explaining the difference between **API**s and **SDK**s:
`https://nordicapis.com/what-is-the-difference-between-an-api-and-an-sdk/`
An API is just an interface between a program (or a server) and another program (or a server). A SDK is a set of development tools to use something.

04.10.19 ~ Introduction to RabbitMQ

I read this intro to **RabbitMQ** and **Queue Managers** in general:
`https://www.cloudamqp.com/blog/2015-05-18-part1-rabbitmq-for-beginners-what-is-rabbitmq.html`

I started reading the code of the channels.py file of tl_dm_core. From what I understand, these `RabbitMQInputChannel` and `RabbitMQOutputChannel` are subclasses of Rasa `InputChannel` and `OutputChannel` which make use of the RabbitMQ queuing management mechanism (implemented in the library `pika`). I need to look at other examples of channels, maybe try to do something with slack first.

06.11.19 ~ Microsoft Common Data Model (CDM)

I read this article presenting Microsoft's **CDM**:
`https://docs.microsoft.com/en-us/common-data-model/`
which from what I understand is a framework to unify different data-base, and have common categories for their content. A kind of extra layer of abstraction which enables simpler development of apps which need to use the data, and also simpler development of data integration procedure (adding data I guess).

15.11.19 ~ Web development resources

I found this page on Full Stack Python which lists many online resource to better understand how internet works and **web development:**
`https://www.fullstackpython.com/web-development.html`

17.11.19 ~ Cross-Origin Resource Sharing

I read this tutorial about Cross-Origin Resource Sharing standard (CORS):
`https://www.codecademy.com/articles/what-is-cors`
which seem to be a way for a website to restrict which other websites can make PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH requests to them.

21.11.19 ~ Context Free Grammar

I looked at the basic example of **Context Free Grammar (CFG)** on wikipedia:
`https://en.wikipedia.org/wiki/Context-free_grammar`
From what I understand a grammar is a pattern matching a set of strings. If we parse a sentence, then we can say if the grammar is activated or not.

04.12.19 ~ Download more songs

I downloaded some new songs with

```
youtube-dl -x --audio-format mp3 -o "%(title)s.%(ext)s" "https://www.youtube.com/watch?v=
```

He last song I downloaded was Everybody's fool of Evanescence.

03.01.20 ~ Proxy server

I learned about **proxy servers** there:
`https://en.wikipedia.org/wiki/Proxy_server`
This was mentioned in the documentation about the sanic library:
`https://sanic.readthedocs.io/en/latest/sanic/config.html`

30.01.20 ~ Dockerfile

I read this post which lists the different kinds of **instructions** appearing in a **Dockerfile**:
`https://www.learnitguide.net/2018/06/dockerfile-explained-with-examples.html`
Then I watched this video of the same website explaining how to **write a Dockerfile**:
`https://www.learnitguide.net/2018/06/write-dockerfile-to-build-own-images.html`

05.01.20 ~ docker-compose.yml

I learned a few things concerning docker-compose files: in the lines

```
ports:
  - "9887:8888"
```

the port exposed is the one on the left. The one on the right is the port of the container itself.

Another thing I learned is that what is listed as volume is a way to make accessible inside the docker something which is in the machine running the container (some files for instance).

12.04.20 ~ Automatic back up

I installed Deja Dup to make automatic back ups of my system but when I press the button "Back up now", I get an error: "Backup failed, couldn't understand duplicity version". I found these two pages:
https://askubuntu.com/questions/826646/backup-failed-could-not-understand-duplicity-vers
and
https://askubuntu.com/questions/826646/backup-failed-could-not-understand-duplicity-vers
but nothing seemed to work. When I do "echo $PATH" I get

/home/aritz/anaconda3/bin:/home/aritz/anaconda3/condabin:/home/aritz/bin:/home/aritz/.lo

so I wonder if this duplicity is really installed in the right place.

I asked a question on Kubuntu forum:
https://www.kubuntuforums.net/showthread.php/76668-Backup-error-Duplicity?
p=435730#post435730

29.08.20 ~ Finding causes of computer slowness, Freeing storage space

I stumbled upon this video which is part of a Google Coursera class about troubleshooting:
https://www.coursera.org/lecture/troubleshooting-debugging-techniques/slow-web-server-zX
It explains possible root causes of slowness of a computer.

I also erased several folders and files in my CS directory which were taking a huge amount of space. The biggest one was the Wheels directory which contained the repo Joffrey was trying to run for SBB. It was 50 GiB large. I need to remove it from my hard drive back up as well.

02.08.20 $\sim$ **Binary representation of negative numbers**

By reading the wikipedia article on p-adic numbers, I discovered this article about **binary representation of negative numbers**:
`https://en.wikipedia.org/wiki/Two%27s_complement`

25.08.2020 $\sim$ **Webscraping**

I discovered the bash command `wget` which can help easily download web pages from internet:
`https://www.linuxjournal.com/content/downloading-entire-web-site-wget`
and here are options to download only html files:
`https://superuser.com/questions/709702/how-to-crawl-using-wget-to-download-only-html-fil`

Once one has the html files, one can use a python library called `beautifulsoup` to remove the html tags:
`https://pypi.org/project/beautifulsoup4/`

08.09.2020 $\sim$ **Rsync to transfer data**

Nikos made me discover the shell utility Rsync. It allows among other possibilities to transfer files from a remote location to a local one. This tutorial is good:
`https://www.digitalocean.com/community/tutorials/how-to-use-rsync-to-sync-local-and-remo`

3 Knowledge I need to gain

3.1 Knowledge in deep learning I need to gain

- I should know all the theoretical parts of **Bert** by heart. Here I will list some specific questions I should be able to answer:
  - How does the decoder of the transformer outputs recursively the predicted tokens? How is this related to mask in Bert?
- I should learn how to use pytorch implementation of **Bert** and similar big models based on transformer.
- I should learn how to use **transfer learning** with DL. This is used with Pytorch in Fast.ai. I would like to be able to implement these semi-supervised tasks like language modeling or auto-encoding (see paper Semi-supervised sequence learning). I could maybe use this approach in DNA classification (project with Cécile Mingard).

- I should learn to use **Torchtext** for NLP. This tutorial:
  `http://anie.me/On-Torchtext/`
  could be a good start.
- I should learn how to use the **learning rate finder** used in Fast.ai.
- I should learn how to efficiently optimize neural networks. Training LSTM seems not to be a trivial task if I believe what is explained p.3 of the paper called "Semi-supervised sequence learning".

3.2 Python libraries I should learn

1. I could learn **flask** which is a web-development framework for python developers which is supposed to be quite easy to use. Roman was using it (to test the nlu service for instance). The official website seems to be a nice place to start:
   `http://flask.pocoo.org/`
2. I could learn **Pandas** since it seems to be very useful for data science.

3.3 Tasks I should perform faster

1. Selecting a text span in the code for copy pasting. There must be some fast way to do it with VIM.
2. Navigating through tabs in mozilla firefox (coming back to the previous one).

3.4 Other things I need to learn

1. (09.10.19) I should learn about **system design**. Rishu pointed me to this set of lectures available on youtube:
   `https://www.youtube.com/watch?v=-W9F__D3oY4`
2. (16.09.19) Aurélien told me about **Software Design Patterns**:
   `https://en.wikipedia.org/wiki/Software_design_pattern`
   I need to know the basic ones.
3. (09.09.19) Nikos found a book and a course of ETH on **Document Retrieval**. That could be an interesting topic and an asset to put on my CV.
4. (11.06.19) I should improve my **general programming skills**. I should read "The Clean Coder" that Aurélien advised me. One can find it online for free here:
   `https://github.com/NileshGule/Ebooks/blob/master/The%20Clean%20Coder%20A%20Code%20of%20Conduct%20for%20Professional%20Programmers.pdf`
5. I could learn how to write a basic **Dockerfile** and a **docker-compose** file. It would be good if I was able to create and run a docker image from scratch.

6. It would be good to learn how to **measure the computational costs** (at training and at testing) of ML models
7. I should improve my basic python skills. leetCode could be a helpful website.
8. I should finish the **regex golf**.
9. I should know a bit of `spark` (I could start here). Mike mentioned that it is used in his job.
10. I should be able to do a bit of `bash` scripting
11. I should learn to use VIM editor and add it to pycharm. For this I should finish VIM adventure.
12. I would like to improve my Software engineer skills. This page gives some hints on how to do it for free:
    WikiHow: How to be a Sofware Engineer for free
    I should also look at the different books used to prepare for google interview ("Cracking the interview" or "Cracking the system design interview").

4 Ideas of things to do

1. (14.01.20) It would be fun to try to implement every architecture presented here in order to do sentiment analysis:
   `https://blog.paralleldots.com/data-science/breakthrough-research-papers-and-models-`
2. (16.12.19) I read in Via that so far it's not possible to generate the shape of snow flakes with computers. I don't know exactly what is meant by this but it would be interesting to look at this problem. Maybe one could use some random fractals.
3. (29.10.19) Something awesome to do would be to have a computer with to which you could communicate by using both the voice and gestures.
4. (22.10.19) I could try to do a notebook for **style transfer**.
5. 03.09.19 It would be good to create a **service** that I would put **into a server inside a docker**, and query it from my computer.
6. 11.06.19 I should read the **master thesis of Aurélien**.
7. 11.06.19 I should look at the lectures of this course on **NLP** from **Stanford**:
   `http://web.stanford.edu/class/cs224n/index.html#schedule`
8. 11.06.19 Aurélien told me about a new programming language called **Rust**:
   `https://www.rust-lang.org/`
   It could be a new tool.

9. 21.11.17 Sid told me that the course Data Mining is very interesting, and that the projects are very interesting.
10. I could maybe read the introduction and the abstract of each chapter of the book Artificial Intelligence of Russel.
11. Maybe I should do all the bash exercises given by TheAlternative
12. 25.01.18 I would like to look at these adversarial neural networks. Yann LeCun was quite enthusiastic about it in his talk.

5 Questions

1. 03.11.17 When do I use Bayesian inference in real life?
2. 06.11.17 If we normalize de data X what should we do with the response variable y? And how do we do prediction on a new sample x_new?

6 LeetCode Problems I looked at

Here is a list of the problem from leetcode.com I looked at:

- 1. Two sums
- 167. Two Sum II - Input array is sorted
- 653. Two Sum IV - Input is a BST
- 7. Reverse Integer
- 190. Reverse Bits
- 490. The Maze
- 3. Longest Substring Without Repeating Characters
- 159. Longest Substring with At Most Two Distinct Characters
- 505. The Maze II
- 695. Max Area of Island
- 38. Count and Say
- 271. Encode and Decode Strings
- 10. Regular Expression Matching
- 6. ZigZag Conversion

- 310. Minimum Height Trees
- 663. Equal Tree Partition
- 563. Binary Tree Tilt
- 101. Symmetric Tree
- 114. Flatten Binary Tree to Linked List
- 94. Binary Tree Inorder Traversal
- 199. Binary Tree Right Side View