



Corso di Laurea Triennale in Informatica

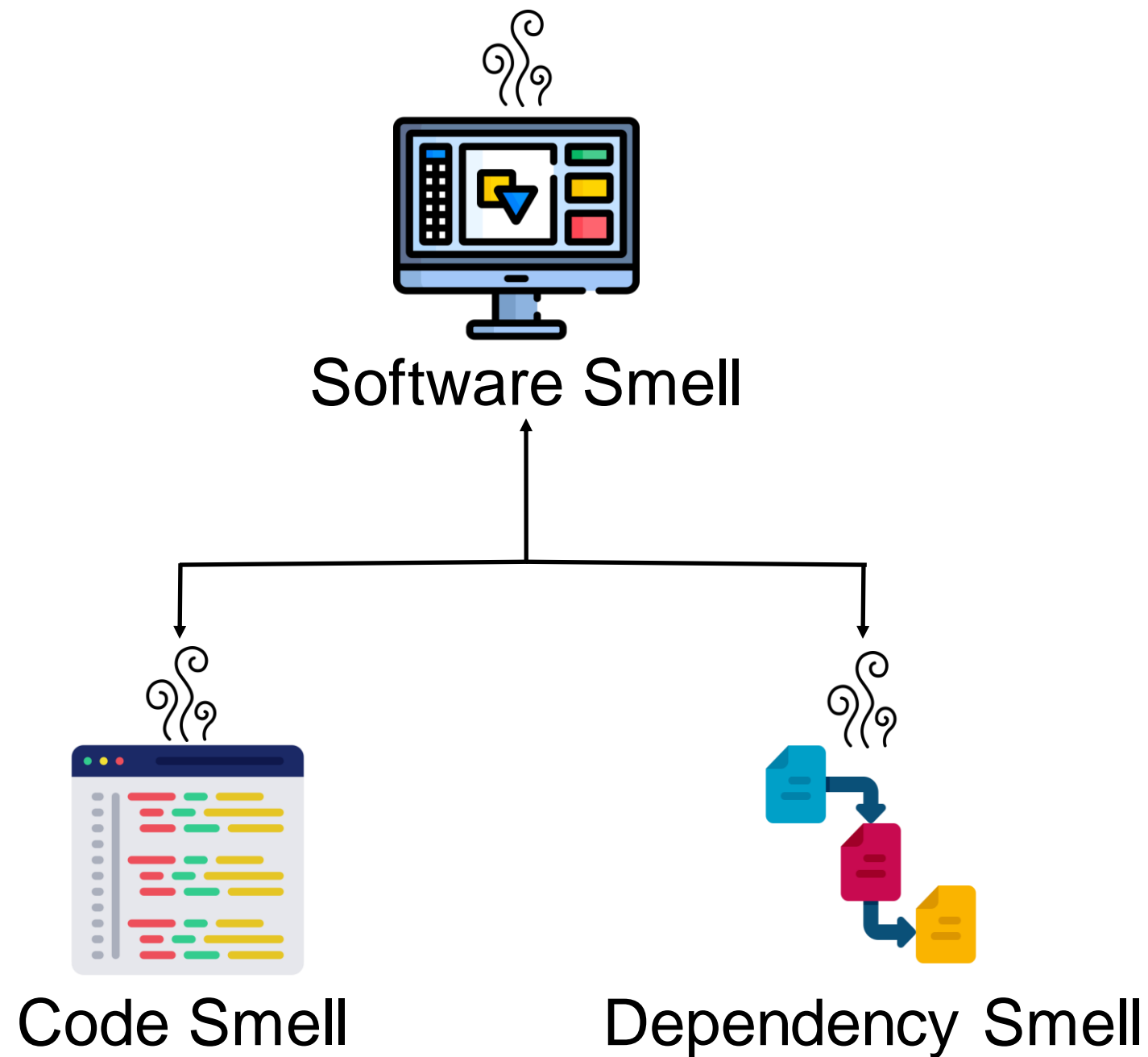
# Identificazione di Bloated Dependency in Python

**Prof. Carmine Gravino**  
**Prof. Fabio Palomba**  
**Dott. Giammaria Giordano**  
**Dott. Emanuele Iannone**

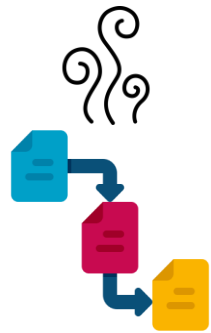
**Daniele Fabiano**  
**Mat.: 0512110373**



# Introduzione e Background



# Introduzione e Background

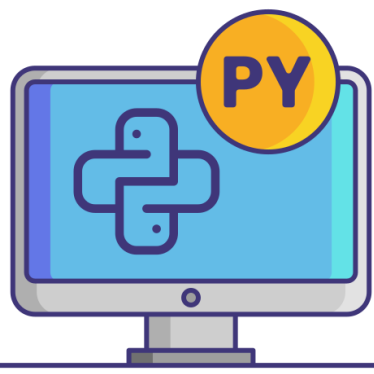


Dependency Smell

Bloated Dependency

```
1 import os
2 from typing import Dict, List, Tuple
3
4 import numpy as np
5 import onnxruntime as ort
6 from PIL import Image
7 from PIL.Image import Image as PILImage
```

base.py



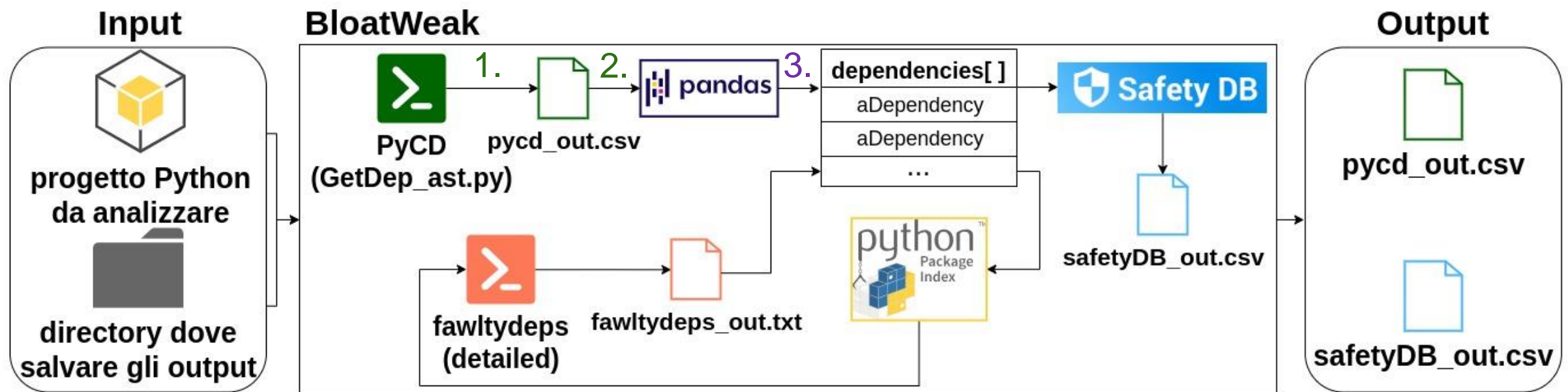
Analisi di progetti Python

```
14 install_requires = [
15     "numpy", ✓
16     "onnxruntime", ✓
17     "scikit-image" ✗
18     "pillow", ✓
19 ]
```

setup.py

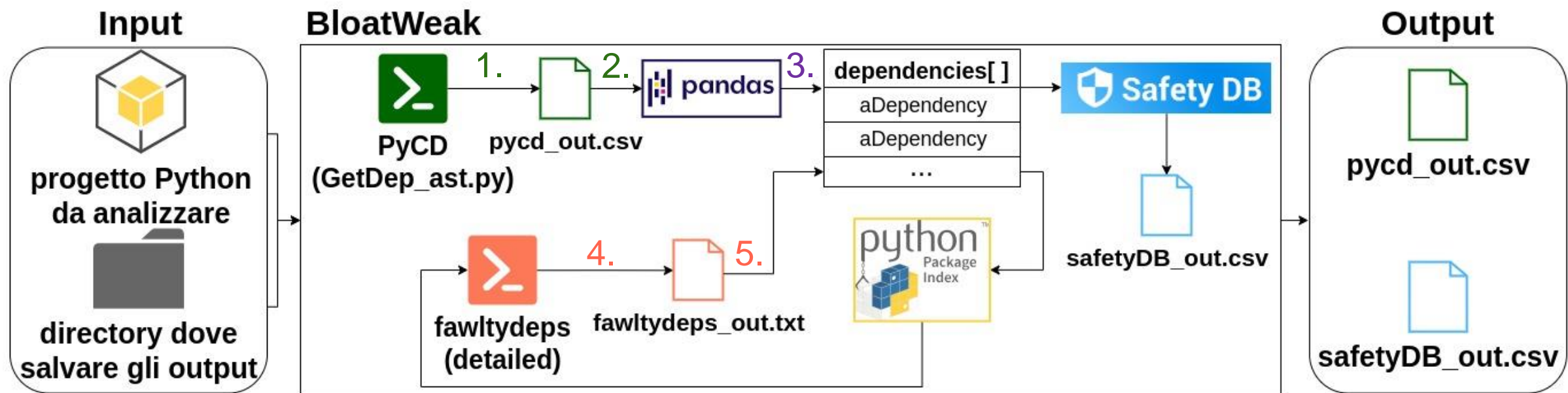


Vulnerabilità nelle dipendenze

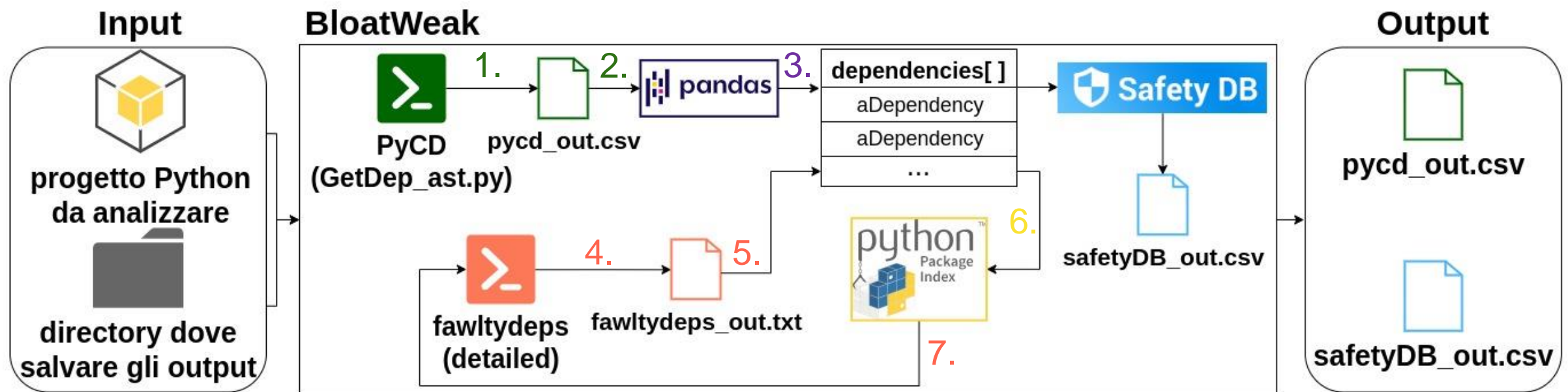


```
$ python3 GetDep_ast.py /home/daniele/git/voltron/ /home/daniele/voltron_dep.csv
```

dep	version	filepath	type	condition	status
cursor	==*	/home/daniele/git/voltron/setup.py	install_requires	@(sys.platform == 'win32')@*	*
scruffington	>=0.3.6	/home/daniele/git/voltron/setup.py	install_requires	*@*	*
flask	==*	/home/daniele/git/voltron/setup.py	install_requires	*@*	*
flask_restful	==*	/home/daniele/git/voltron/setup.py	install_requires	*@*	*
blessed	==*	/home/daniele/git/voltron/setup.py	install_requires	*@*	*
pygments	==*	/home/daniele/git/voltron/setup.py	install_requires	*@*	*
requests	==*	/home/daniele/git/voltron/setup.py	install_requires	*@*	*
requests_unixsocket	==*	/home/daniele/git/voltron/setup.py	install_requires	*@*	*
six	==*	/home/daniele/git/voltron/setup.py	install_requires	*@*	*
pysigset	==*	/home/daniele/git/voltron/setup.py	install_requires	*@*	*



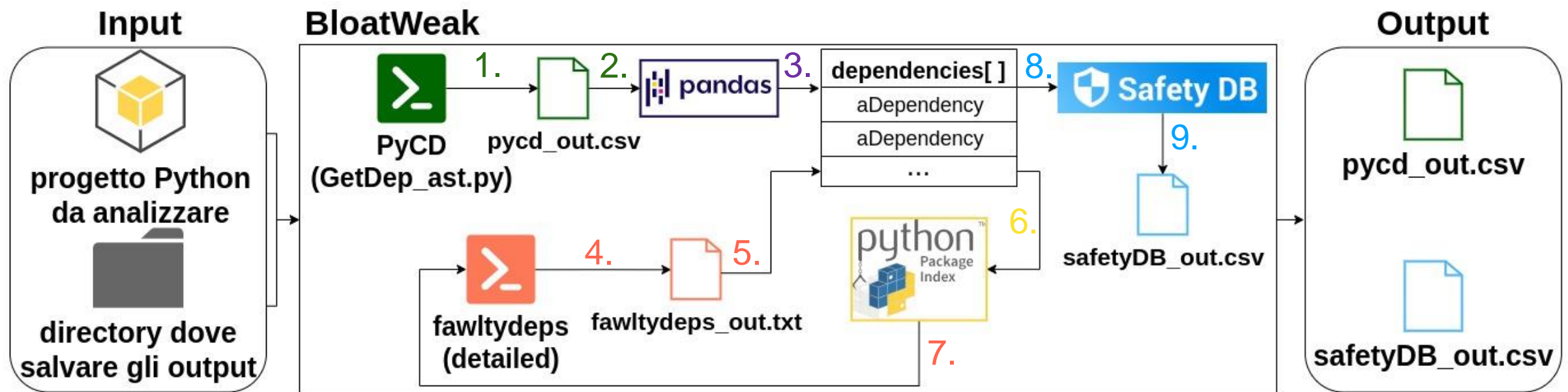
```
daniele@thinkbook14:~  
[daniele@thinkbook14 ~]$ fawltydeps /home/daniele/git/voltron/ --check-unused --detailed  
  
These dependencies appear to be unused (i.e. not imported):  
- 'flask_restful' declared in:  
  /home/daniele/git/voltron/setup.py  
- 'pysigset' declared in:  
  /home/daniele/git/voltron/setup.py  
- 'scruffington' declared in:  
  /home/daniele/git/voltron/setup.py  
[daniele@thinkbook14 ~]$
```





```
daniele@thinkbook14:~/git/BloatWeak/bloatWeak
(xenv) [daniele@thinkbook14 bloatWeak]$ fawltydeps /home/daniele/git/voltron/ --check-unused --detailed

These dependencies appear to be unused (i.e. not imported):
- 'flask_restful' declared in:
  /home/daniele/git/voltron/setup.py
- 'pysigset' declared in:
  /home/daniele/git/voltron/setup.py
(xenv) [daniele@thinkbook14 bloatWeak]$
```



```
daniele@thinkbook14:~/git/BloatWeak/bloatWeak
(xenv) [daniele@thinkbook14 bloatWeak]$ python3 bloatWeak.py /home/daniele/git/voltron/ /home/daniele/voltron/
PyCD output obtained in csv format (pycd_out.csv)
Checking unused dependencies with FawltyDeps
Installing unused dependencies
SafetyDB filtered output obtained in csv format (safetyDB_out.csv)

Bloated dependencies for your project:
flask_restful==* declared in: /home/daniele/git/voltron/setup.py
pysigset==* declared in: /home/daniele/git/voltron/setup.py

You can find all the detailed output in the files stored in /home/daniele/voltron/ directory
(xenv) [daniele@thinkbook14 bloatWeak]$
```



Analisi del codice sorgente e calcolo delle metriche

10 progetti analizzati

4 Veri Positivi (VP)

2 Veri Negativi (VN)

4 Falsi Positivi (FP)

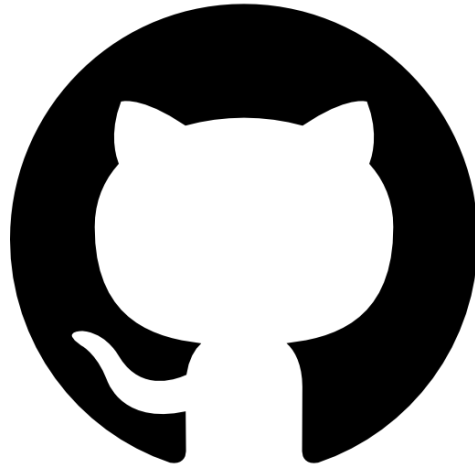
0 Falsi Negativi (FN)

$$\text{Precision} = \frac{VP}{VP+FP} = 0.5$$

$$\text{Recall} = \frac{VP}{VP+FN} = 1$$

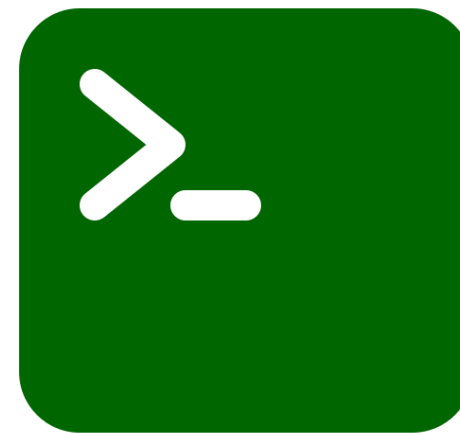
$$\text{Accuracy} = \frac{VP+VN}{VP+VN+FP+FN} = 0.6$$

1.



Clone del progetto

2.



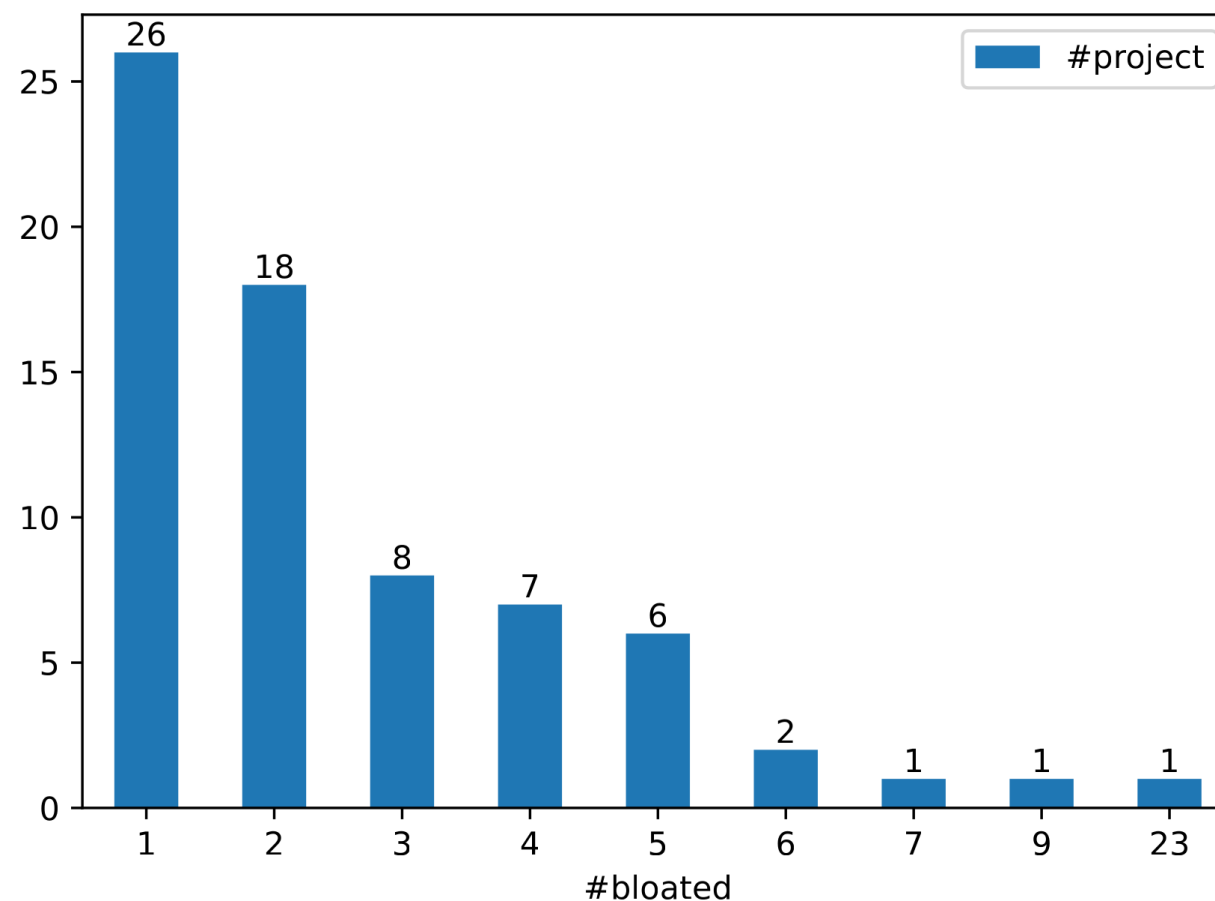
Esecuzione di BloatWeak

3.



Analisi del numero delle bloated dependency e delle vulnerabilità

**Q RQ<sub>1</sub>.** *Quanto è diffuso il fenomeno delle bloated dependency, all'interno di progetti Python che fanno uso di soluzioni ML?*



70/365 progetti affetti dal fenomeno delle bloated dependency

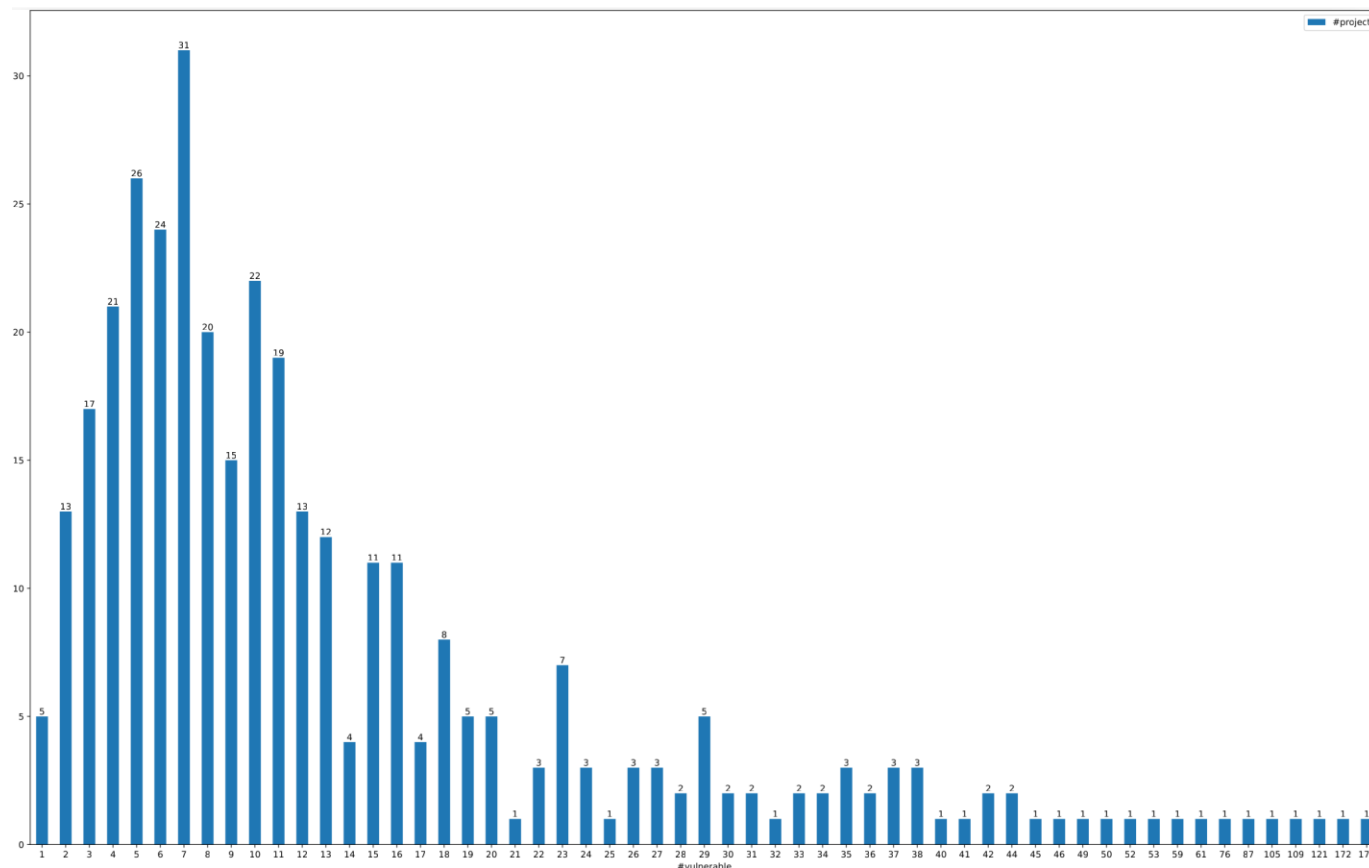
Media: 8 bloated dependency

Moda: 1 bloated dependency

Mediana: 5 bloated dependency

Correlazione numero dipendenze totali e numero bloated dependency

**Q RQ<sub>2</sub>.** *Quanto è diffuso il fenomeno delle dipendenze vulnerabili, all'intero di progetti Python che fanno uso di soluzioni ML?*



355/365 progetti affetti dal fenomeno delle dipendenze vulnerabili

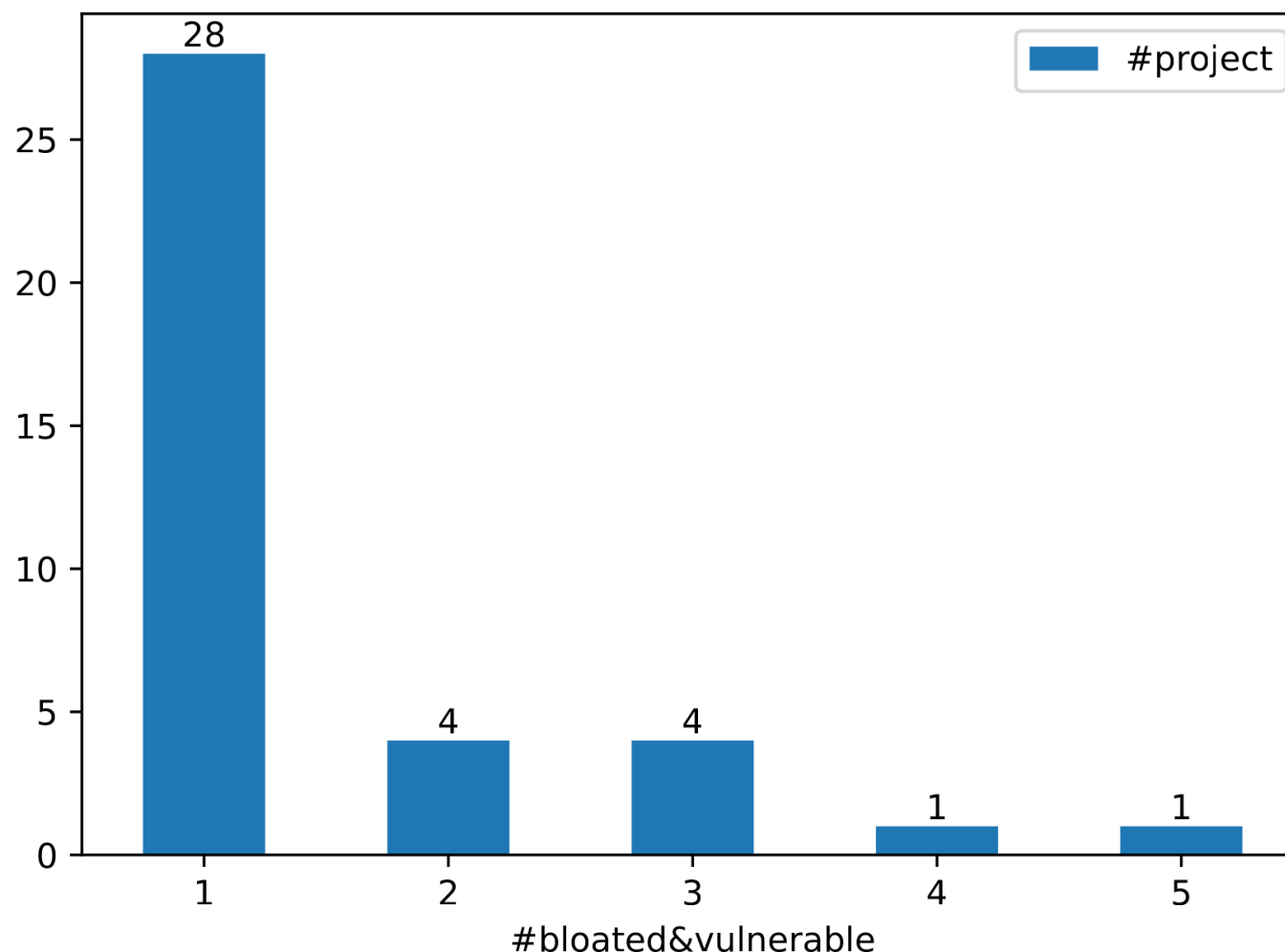
Media: 38 dipendenze vulnerabili

Moda: 7 dipendenze vulnerabili

Mediana: 29 dipendenze vulnerabili

Correlazione numero dipendenze vulnerabili e numero vulnerabilità

**Q RQ<sub>3</sub>.** *Come sono legate tra loro le diffusioni dei fenomeni delle bloated dependency e delle dipendenze vulnerabili, all'interno di progetti Python che fanno uso di soluzioni ML?*



38/70 progetti affetti da entrambi i fenomeni

Media: 3 dipendenze bloated e vulnerabili

Moda: 1 dipendenza bloated e vulnerabile

Mediana: 3 dipendenze bloated e vulnerabili

Gravità della combinazione dei due fenomeni



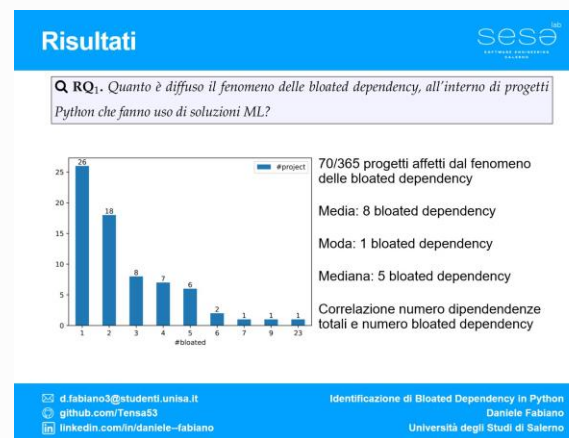
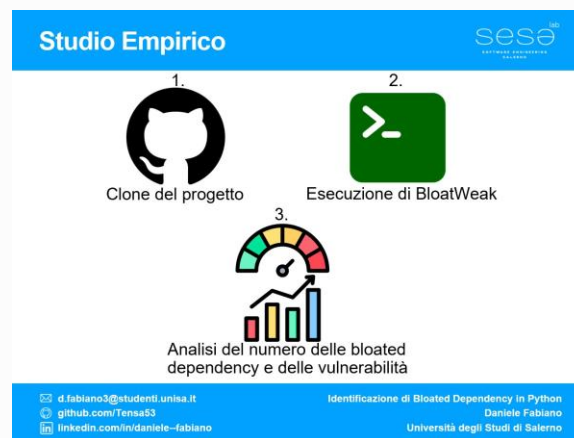
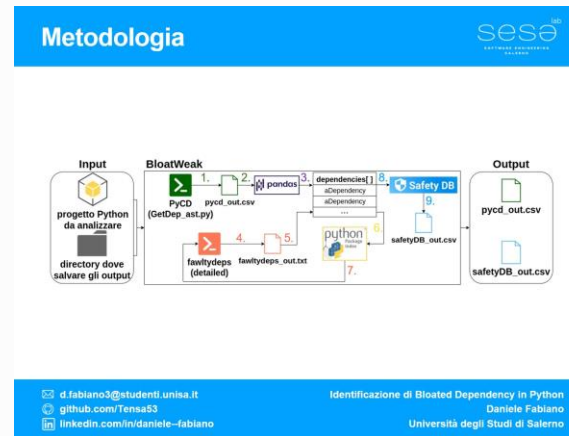
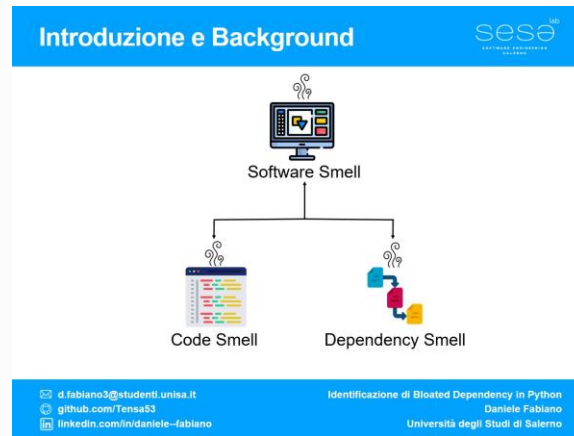
## Limitazioni Attuali

- Dataset analizzato
- Limiti dei tool interni
- Valore semantico delle dichiarazioni
- Errori nell'esecuzione

## Sviluppi Futuri

- Analisi del funzionamento dei gestori pacchetti
- Analisi delle "good and bad practises" degli sviluppatori
- Analisi approfondita della pratica del riuso

# Identificazione di Bloating Dependency in Python



Grazie!



Questa tesi ha contribuito a piantare un albero in Camerun



Daniele Fabiano

d.fabiano3@studenti.unisa.it

github.com/Tensa53

linkedin.com/in/daniele--fabiano