

# MODULAR CODE FOR DATA SCIENTIST

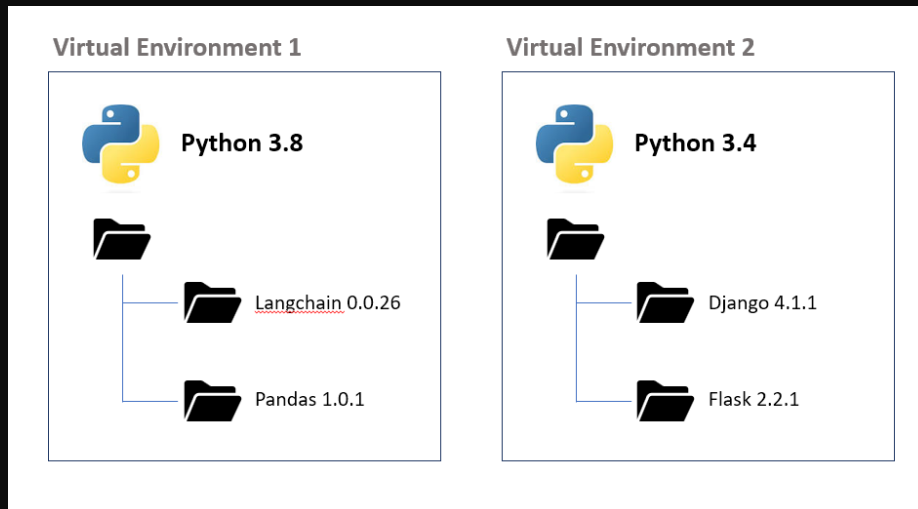
A weekend project

# AGENDA

1. Virtual environment
  - Kenapa perlu virtual environment?
2. Modular Code
  - Jupyter Notebook vs Modular
3. Practical Implication

# Virtual Environment

*: kamu punya sebuah laptop di dalam laptopmu dengan OS yang beda, program yang beda*



contoh virtual environment:

- Anaconda
- Pyenv
- Venv

# Kenapa nulis semua kodingan di jupyter itu praktek yang buruk?

1. Code Reusability

*you can't reuse the code from one notebook inside another notebook*

2. Unorganized

*terkadang, kodingan di notebook ga berurutan. Ini bikin masalah ketika orang lain baca notebookmu dan bikin kita sulit ngedebug*

3. Not ideal for production

*kita harus menulis ulang kodingan ke file .py jika ingin menaikkan kodingan kita ke production*

Sedangkan, dengan menerapkan modularity kita cuma perlu 8 block notebook untuk dapat hasil AUC

```
▶ print(roc_auc_score(y_test, automl.predict_proba(X_test)[:, 1]))
```

[8] ✓ 0.0s Python

... 0.9520754451255935

[https://github.com/andrioktavianto/modular\\_code\\_data\\_science](https://github.com/andrioktavianto/modular_code_data_science)

# Practical Implication

1. Modularity meningkatkan produktivitas kita karena kita ga perlu nulis ulang kodingan
2. Modularity memudahkan kita untuk debug kodingan
3. Kalo mau deploy ke production gampangg

## Next Improvement

1. Masih ada function yang belum menerapkan prinsip SOLID, seperti fungsi `clean_data()`
2. Penerapan class dan OOP
3. Bikin fungsi untuk EDA, cross validation, dan confusion matrix