# Byte-compiled / optimized / DLL files

\_\_pycache\_\_/

\*.py[cod]

\*$py.class

# C extensions

\*.so

# Distribution / packaging

.Python

build/

develop-eggs/

dist/

downloads/

eggs/

.eggs/

lib/

lib64/

parts/

sdist/

var/

wheels/

share/python-wheels/

\*.egg-info/

.installed.cfg

\*.egg

MANIFEST

# PyInstaller

# Usually these files are written by a python script from a template

# before PyInstaller builds the exe, so as to inject date/other infos into it.

\*.manifest

\*.spec

# Installer logs

pip-log.txt

pip-delete-this-directory.txt

# Unit test / coverage reports

htmlcov/

.tox/

.nox/

.coverage

.coverage.\*

.cache

nosetests.xml

coverage.xml

\*.cover

\*.py,cover

.hypothesis/

.pytest\_cache/

cover/

# Translations

\*.mo

\*.pot

# Django stuff:

\*.log

local\_settings.py

db.sqlite3

db.sqlite3-journal

# Flask stuff:

instance/

.webassets-cache

# Scrapy stuff:

.scrapy

# Sphinx documentation

docs/\_build/

# PyBuilder

.pybuilder/

target/

# Jupyter Notebook

.ipynb\_checkpoints

# IPython

profile\_default/

ipython\_config.py

# pyenv

# For a library or package, you might want to ignore these files since the code is

# intended to run in multiple environments; otherwise, check them in:

# .python-version

# pipenv

# According to pypa/pipenv#598, it is recommended to include Pipfile.lock in version control.

# However, in case of collaboration, if having platform-specific dependencies or dependencies

# having no cross-platform support, pipenv may install dependencies that don't work, or not

# install all needed dependencies.

#Pipfile.lock

# UV

# Similar to Pipfile.lock, it is generally recommended to include uv.lock in version control.

# This is especially recommended for binary packages to ensure reproducibility, and is more

# commonly ignored for libraries.

#uv.lock

# poetry

# Similar to Pipfile.lock, it is generally recommended to include poetry.lock in version control.

# This is especially recommended for binary packages to ensure reproducibility, and is more

# commonly ignored for libraries.

# https://python-poetry.org/docs/basic-usage/#commit-your-poetrylock-file-to-version-control

#poetry.lock

# pdm

# Similar to Pipfile.lock, it is generally recommended to include pdm.lock in version control.

#pdm.lock

# pdm stores project-wide configurations in .pdm.toml, but it is recommended to not include it

# in version control.

# https://pdm.fming.dev/latest/usage/project/#working-with-version-control

.pdm.toml

.pdm-python

.pdm-build/

# PEP 582; used by e.g. github.com/David-OConnor/pyflow and github.com/pdm-project/pdm

\_\_pypackages\_\_/

# Celery stuff

celerybeat-schedule

celerybeat.pid

# SageMath parsed files

\*.sage.py

# Environments

.env

.venv

env/

venv/

ENV/

env.bak/

venv.bak/

# Spyder project settings

.spyderproject

.spyproject

# Rope project settings

.ropeproject

# mkdocs documentation

/site

# mypy

.mypy\_cache/

.dmypy.json

dmypy.json

# Pyre type checker

.pyre/

# pytype static type analyzer

.pytype/

# Cython debug symbols

cython\_debug/

# PyCharm

# JetBrains specific template is maintained in a separate JetBrains.gitignore that can

# be found at https://github.com/github/gitignore/blob/main/Global/JetBrains.gitignore

# and can be added to the global gitignore or merged into this file. For a more nuclear

# option (not recommended) you can uncomment the following to ignore the entire idea folder.

#.idea/

# Ruff stuff:

.ruff\_cache/

# PyPI configuration file

.pypirc