

Week03 学习笔记

1. 在自己的终端 (比如 Git Bash、Zsh 等) 配置好 Conda Init, 使得启动终端后, 在提示符 (比如 \$、%) 前能够看到 (base)
2. 使用 conda info 命令查看本机 Conda 的配置信息

```
ttx58@ttx MINGW64 ~
$ conda info

      active environment : None
      user config file   : C:\Users\ttx58\.condarc
populated config files  : C:\Users\ttx58\anaconda3\.condarc
      conda version      : 24.9.2
      conda-build version: 24.9.0
      python version     : 3.12.7.final.0
      solver             : libmamba (default)
virtual packages        : __archspec=1=icelake
                        __conda=24.9.2=0
                        __win=0=0
base environment        : C:\Users\ttx58\anaconda3 (writable)
conda av data dir       : C:\Users\ttx58\anaconda3\etc\conda
conda av metadata url   : None
channel URLs            : https://repo.anaconda.com/pkgs/main/win-64
                        https://repo.anaconda.com/pkgs/main/noarch
                        https://repo.anaconda.com/pkgs/r/win-64
                        https://repo.anaconda.com/pkgs/r/noarch
                        https://repo.anaconda.com/pkgs/msys2/win-64
                        https://repo.anaconda.com/pkgs/msys2/noarch
```

3. 使用 conda env list 命令查看已有的 Conda 环境的名称和路径, 理解 Conda 环境的概念

```
ttx58@ttx MINGW64 ~
$ conda env list
# conda environments:
#
base                                C:\Users\ttx58\anaconda3
```

4. 使用 conda create 命令创建两个 Conda 环境, 一个里面安装 Python 3.12 和 requests 软件包, 另一个里面安装 Python 3.9、pandas 和 statsmodels 软件包, 能够在终端里切换 Conda 环境, 验证 Python 和软件包的版本

- 创建环境:

使用 `conda create` 命令来创建一个名为 `py312_env` 的环境, 其中包含 Python 3.12 和 `requests` 软件包。

```
Bash 深色版本 ▾ | ❏
conda create -n py312_env python=3.12 requests
```

- 激活环境:

使用 `conda activate` 命令来激活这个环境。

```
Bash 深色版本 ▾ | ❏
conda activate py312_env
```

- 验证 Python 和软件包版本:

激活后, 可以通过以下命令来验证 Python 和 `requests` 的版本。

5. 使用 conda list 命令显示 Conda 环境里的软件包列表及其版本信息

6. 使用 conda install 命令往 Conda 环境里安装更多的软件包，并验证版本

```
qiang@gqm3win CLANGARM64 ~
$ conda activate prj2
(prj2)
qiang@gqm3win CLANGARM64 ~
$ which python
/c/Users/qiang/anaconda3/envs/prj2/python
(prj2)
qiang@gqm3win CLANGARM64 ~
$ python --version
Python 3.9.21
(prj2)
qiang@gqm3win CLANGARM64 ~
$ python
Python 3.9.21 (main, Dec 11 2024, 16:35:24) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import requests
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'requests'
>>> import pandas
>>> pandas.__file__
'C:\\Users\\qiang\\anaconda3\\envs\\prj2\\lib\\site-packages\\pandas\\__init__.py'
>>> █
```

7. 根据 [文档](#)，配置 Anaconda 清华镜像，加快 conda install 安装软件包的速度，将 conda-forge 设置为默认 Channel，让 conda install 能够安装更多的软件包
- 使用方法

镜像站提供了 Anaconda 仓库与第三方源（conda-forge、msys2、pytorch 等，各镜像站镜像的第三方源并不相同，可以参考下方「第三方镜像源」一节）的镜像，各系统都可以通过修改用户目录下的 .condarc 文件来使用镜像站。

不同系统下的 .condarc 目录如下：

- Linux: \${HOME}/.condarc
- macOS: \${HOME}/.condarc
- Windows: C:\Users\<YourUserName>\.condarc

注：

- * Windows 用户无法直接创建名为 .condarc 的文件，可先执行 conda config --set show_channel_urls yes 生成该文件之后再修改。
- * 由于更新过快难以同步，TUNA 等镜像站不同步 pytorch-nightly，pytorch-nightly-cpu，ignite-nightly 这三个包。
- * 如果您正在从某一镜像源切换到另一镜像源，请检查镜像源是否同步了您所需要的 repo，以及该 repo 是否支持您使用的平台（e.g. linux-64）。
- * 为了保证以下配置在所有镜像站可用，配置中只加入了少量必须的第三方源，您可以在下方的列表中自行寻找并添加其他第三方源。

channels:

- defaults

show_channel_urls: true

default_channels:

- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r

- <https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/mSYS2>
custom_channels:
conda-forge: <https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud>
pytorch: <https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud>
即可添加 Anaconda Python 免费仓库。

使用下列命令清除索引缓存，并安装常用包测试一下。

```
conda clean -i  
conda create -n myenv numpy
```

8. 使用 `pip install` 命令往 Conda 环境里安装 Python 软件包，并验证版本
9. 根据 [文档](#) 配置 PyPI 清华镜像，加快 `pip install` 安装软件包的速度
10. 能够导出 `environment.yml` Conda 环境配置文件，能够删除 Conda 环境，能够用 `environment.yml` 配置文件重建 Conda 环境
11. 理解 Conda 与 Python 的关系，理解 Conda-Forge 与 Conda 的关系，理解 Python 解释器、第三方软件包、PyPI 软件仓库、以及程序/软件包的路径问题
12. 按照 [教程](#) 创建项目目录，在 VS Code 文本编辑器里安装一些支持 Python 开发的常用扩展，编写 `main.py` 脚本，创建该项目专用的 Conda 环境，在终端里激活该环境并成功运行该脚本

主要讲述了如何创建一个本地项目文件夹，并激活环境进行项目开发。首先，在电脑上创建一个隔离的文件夹，避免大写字母、空格等特殊符号。然后，安装 Python 3.12 版本，以便使用第三方软件包。接下来，激活环境并运行脚本，进行静态代码检查和格式化。在配置文件中，可以设置工作空间主题和用户级别设置。此外，还可以通过网络下载 CSV 文件，并使用 Pandas 进行数据处理。最后，提交学习笔记和报告，以便更好地理解 and 掌握项目开发方法。

13. 去 [第 03 周打卡](#) 仓库，按照「操作步骤」说明，用 Fork-PR 方法提交本周学习报告