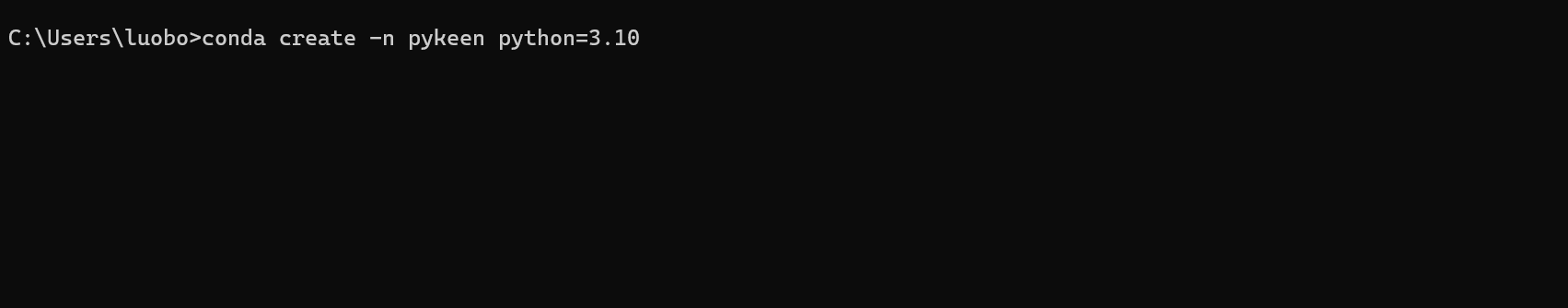
**实验：使用PyKEEN和Neo4j补全知识图谱**

1. 配置conda环境
2. 打开命令行，创建conda环境



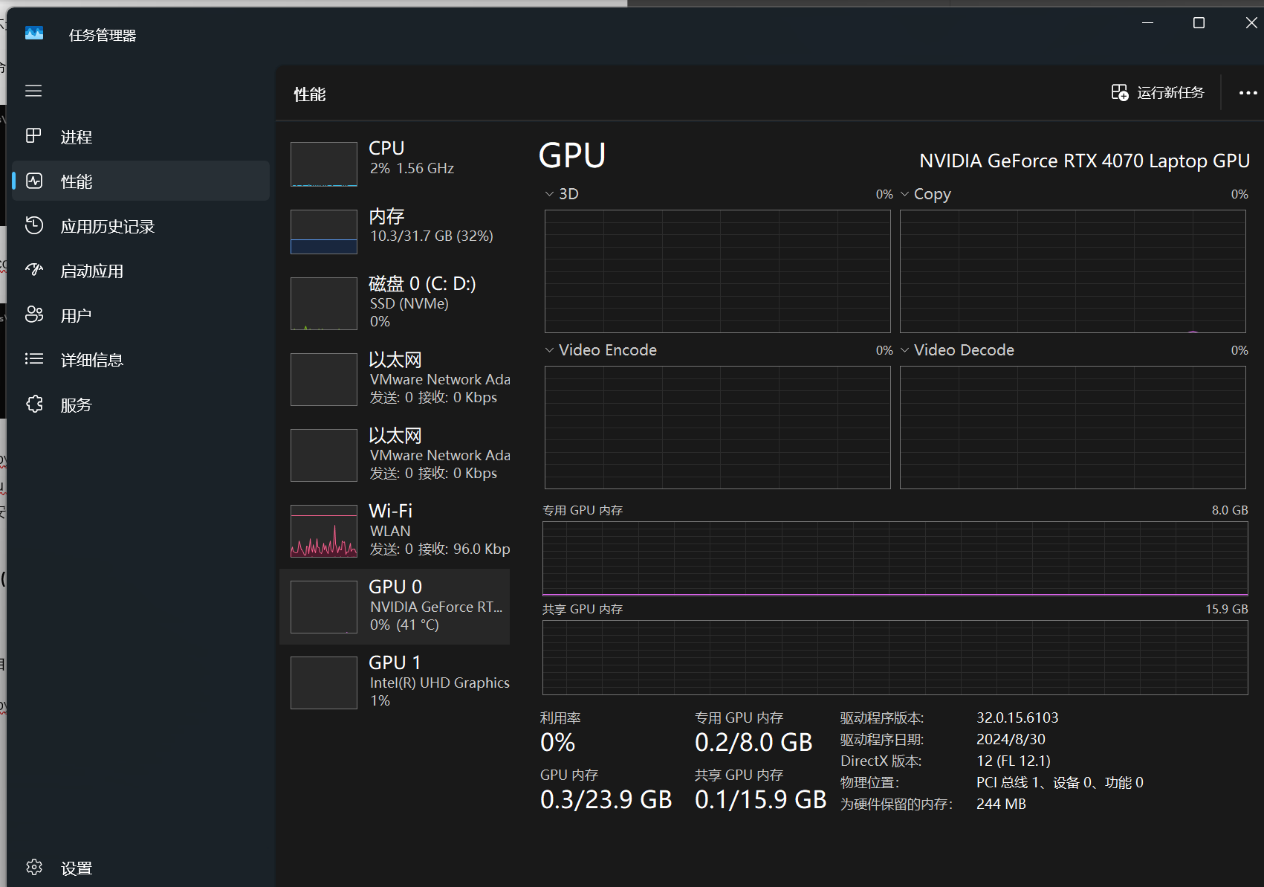
1. 激活conda环境



1. 安装pytorch（推荐，若使用cpu版torch则无需此步骤，pykeen会自动安装cpu版torch，故若使用gpu版torch则需要在安装pykeen前或安装后重新安装gpu版pytorch）

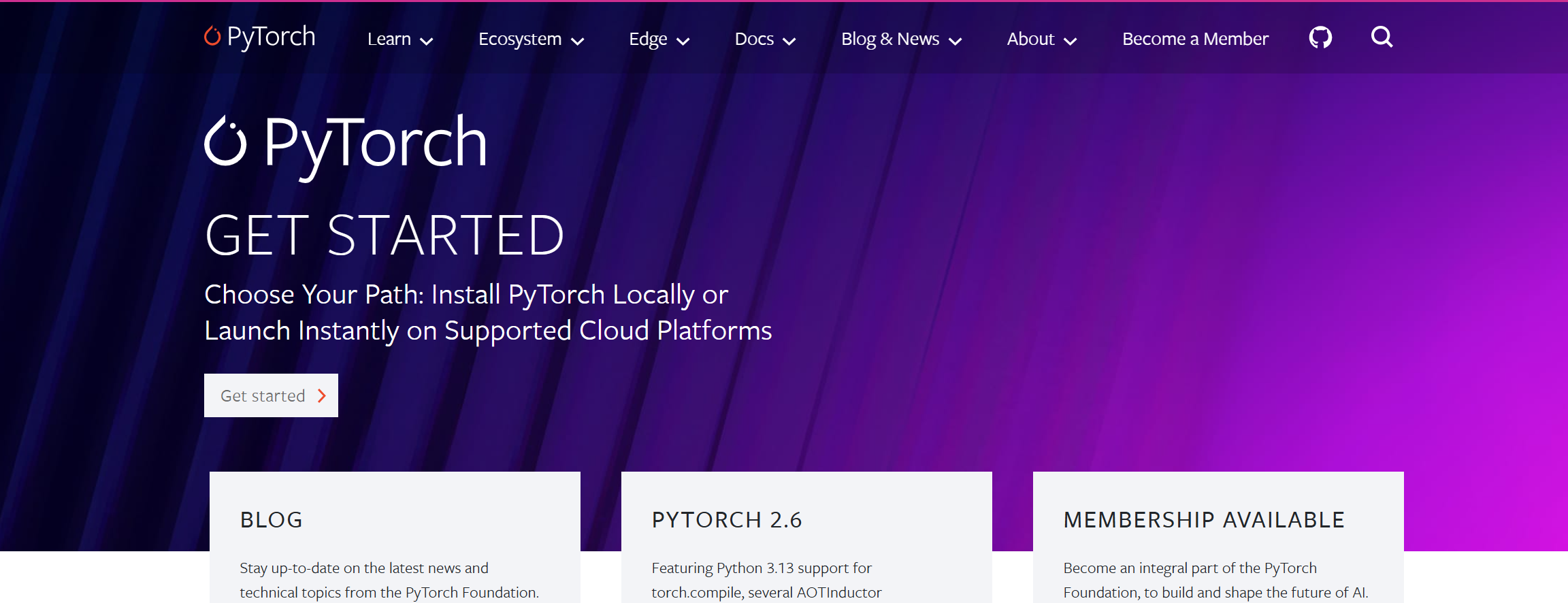
**注意：请检查自己的机器是否配备GPU和GPU的显存，若没有GPU或总显存（专用+共享）<14G则安装CPU版pytorch，无需执行本步骤。**

**查看方法：在任务栏右键，打开任务管理器，切换到性能选项卡，可以看到GPU和显存信息。**

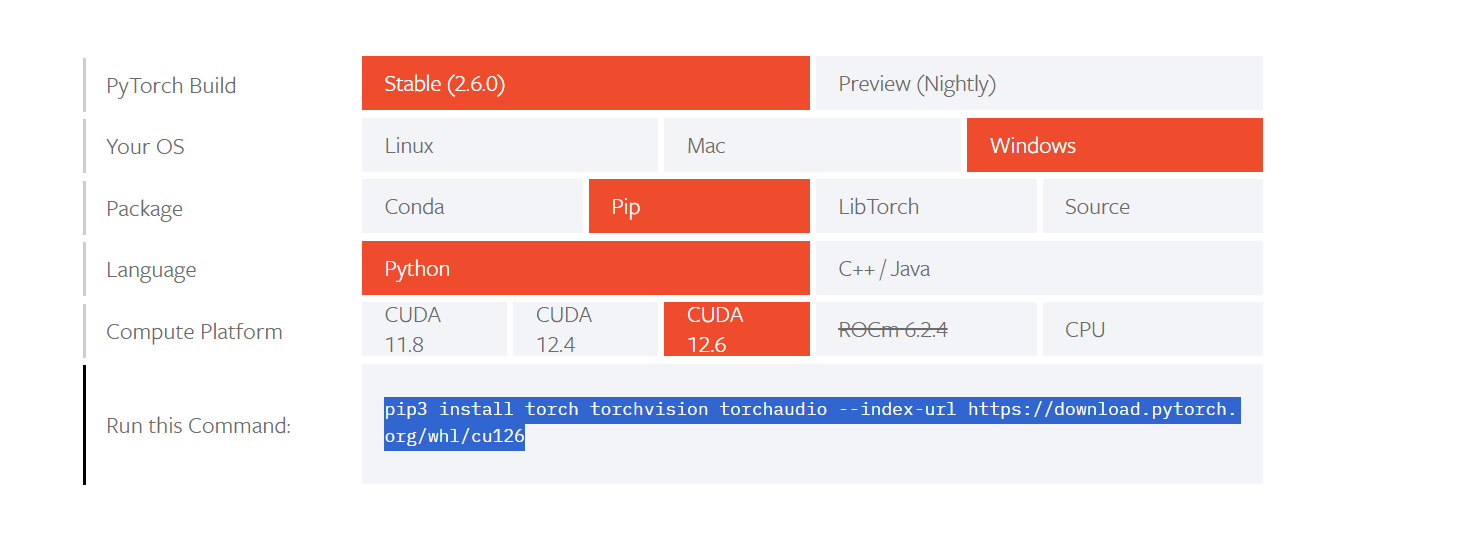


根据自身cuda版本进行安装，此处以cuda12.6为例。

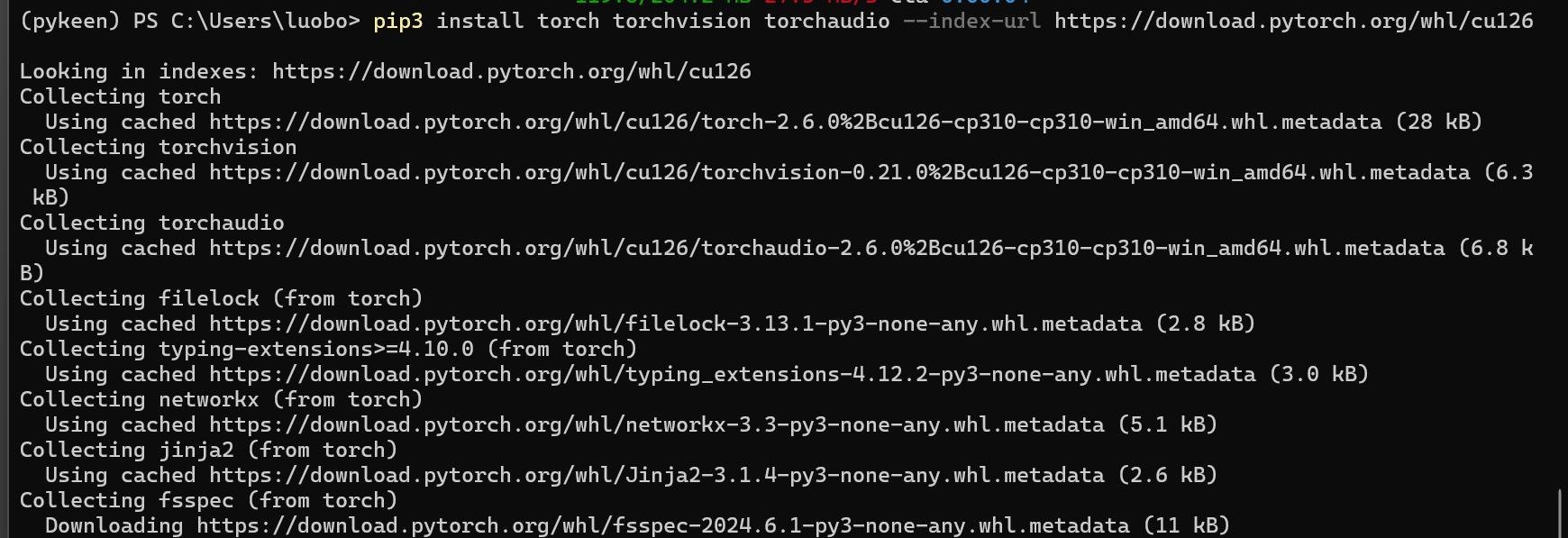
打开pytorch官网，点击get started。



选择合适的版本。

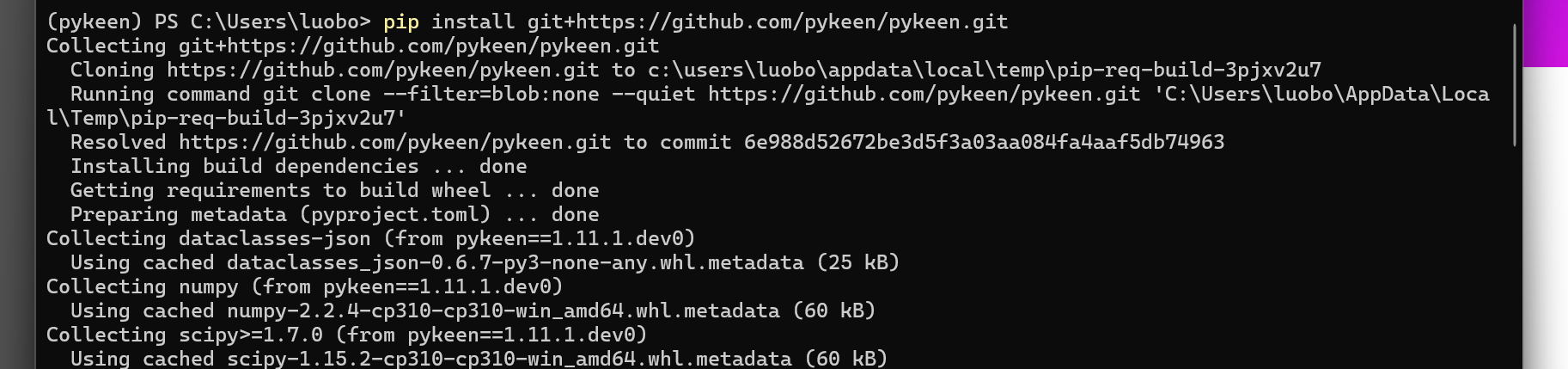


复制下方的命令并运行，等待安装完成。

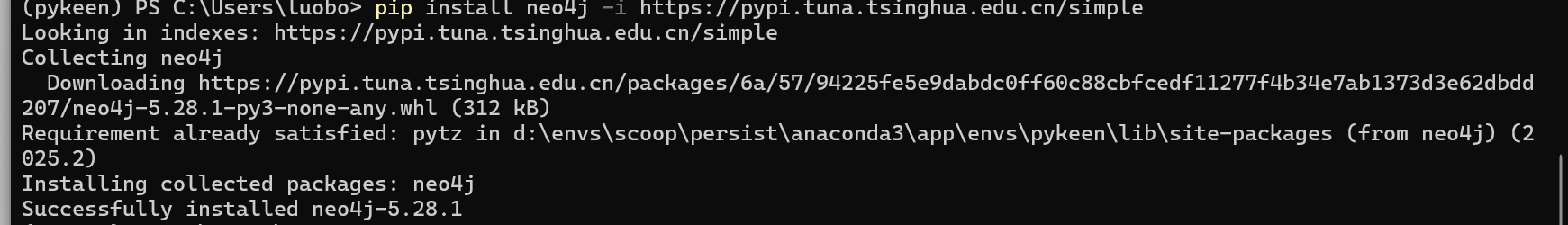


1. 安装pykeen和neo4j驱动库。

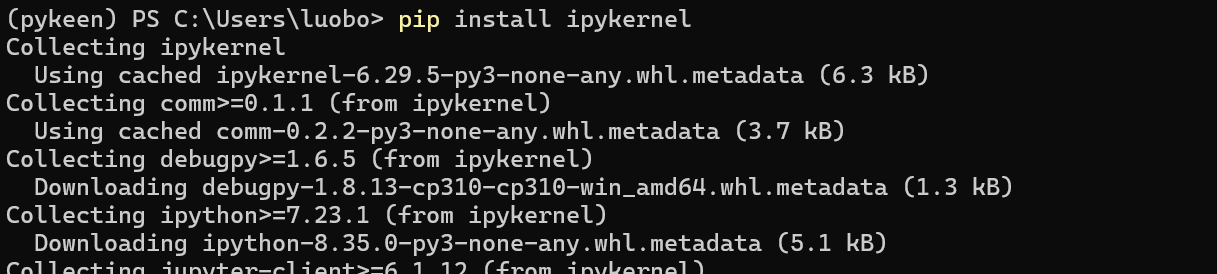
安装最新版pykeen（1.11版本目前尚不支持pytorch2.6，因此需要安装最新提交版本）。



安装neo4j驱动库。



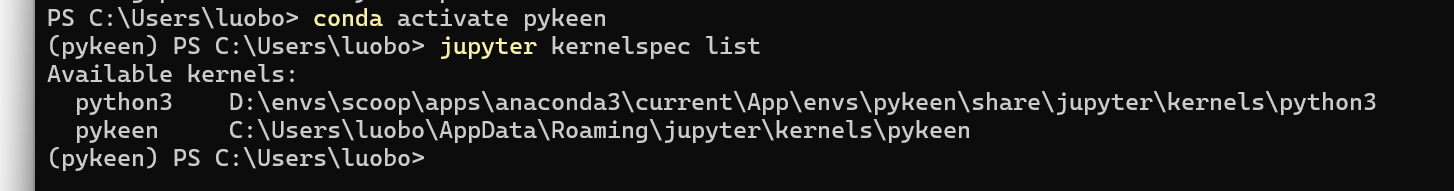
1. 将conda配置到notebook



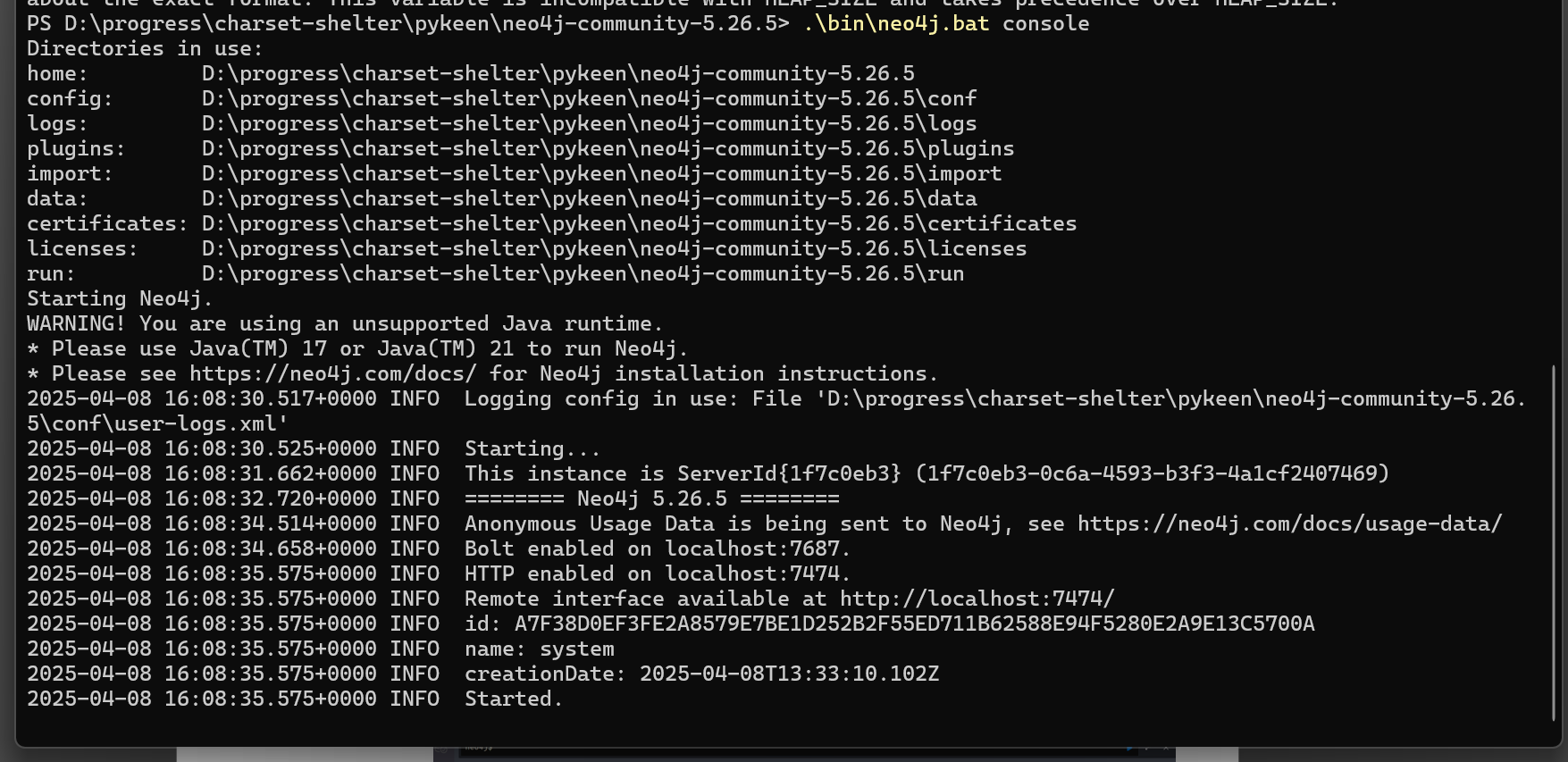
把环境作为内核注册到notebook



1. 检查配置是否成功，出现pykeen即为成功。



1. 启动neo4j
2. 使用实验材料中的neo4j目录（已配置好环境并已导入好数据），启动neo4j。

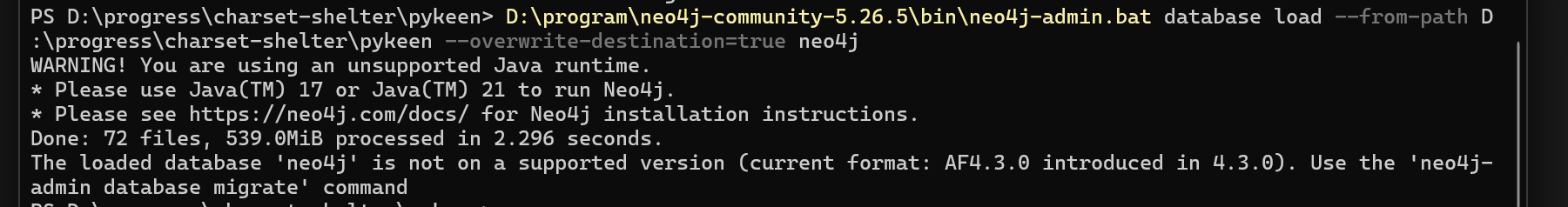


看到”Started”证明neo4j启动成功。

如果想使用已有的neo4j，则需要进行数据导入，执行下列步骤

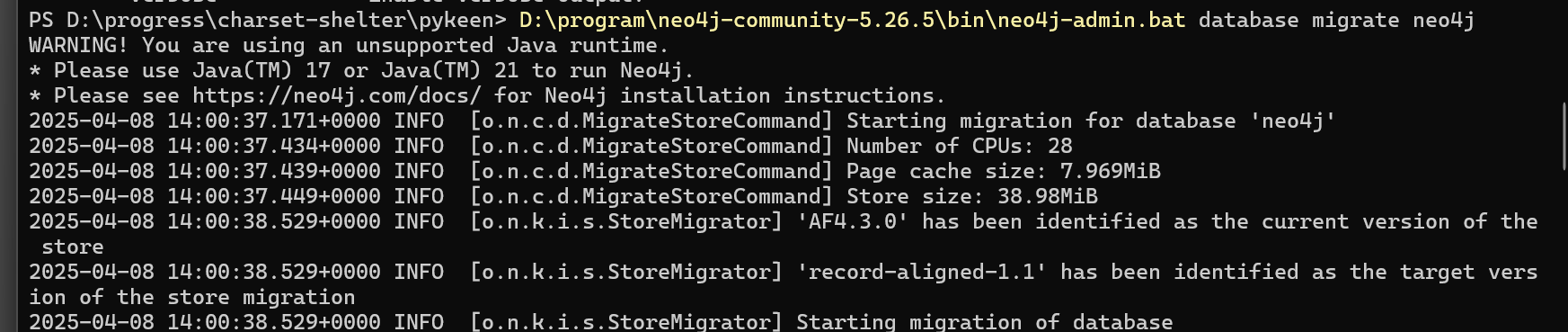
为兼容性起见，建议使用neo4j 5.26.5版本。

导入dump文件：



neo4j-admin指向安装neo4j的目录，from-path为存放dump文件的目录，dump文件名为neo4j.dump

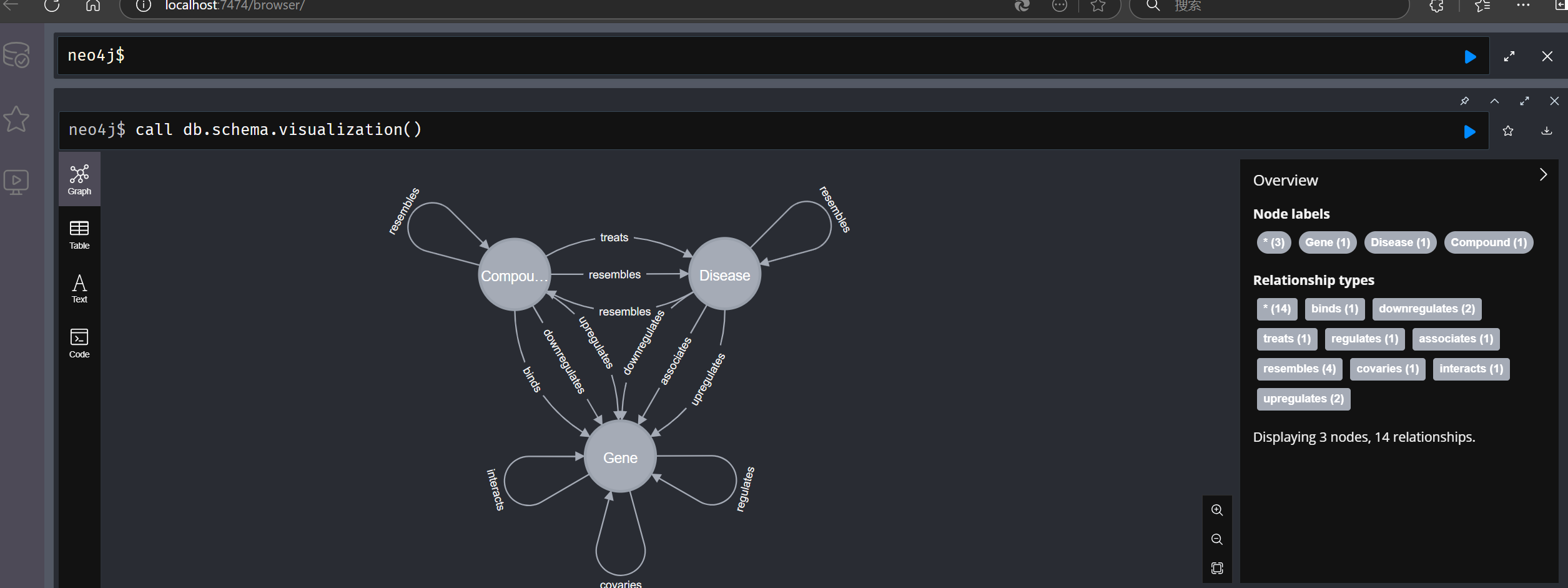
版本迁移：



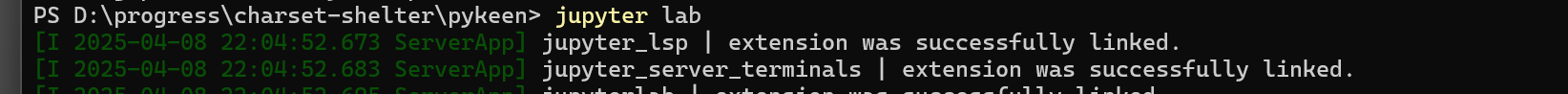
启动neo4j

1. 打开浏览器，初始用户名neo4j，初始密码neo4j，第一次登录强制要求修改密码，此处修改为12345678（最少8位）。

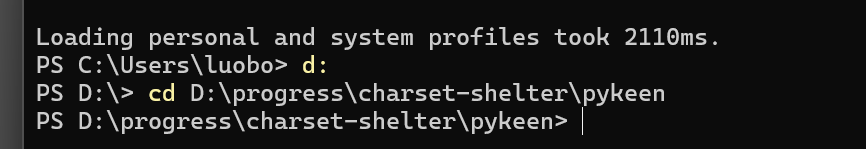
（3）打开浏览器，登录neo4j，用户名neo4j，密码12345678。执行如下语句，观察图谱schema。



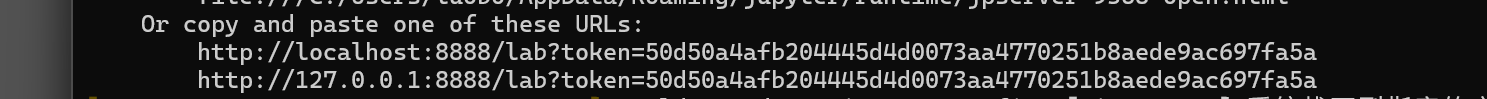
1. 在VS Code中使用Jupyter Notebook
2. 定位至pykeen-neo4j.ipynb所在目录，右键打开菜单，选择“在终端中打开”。



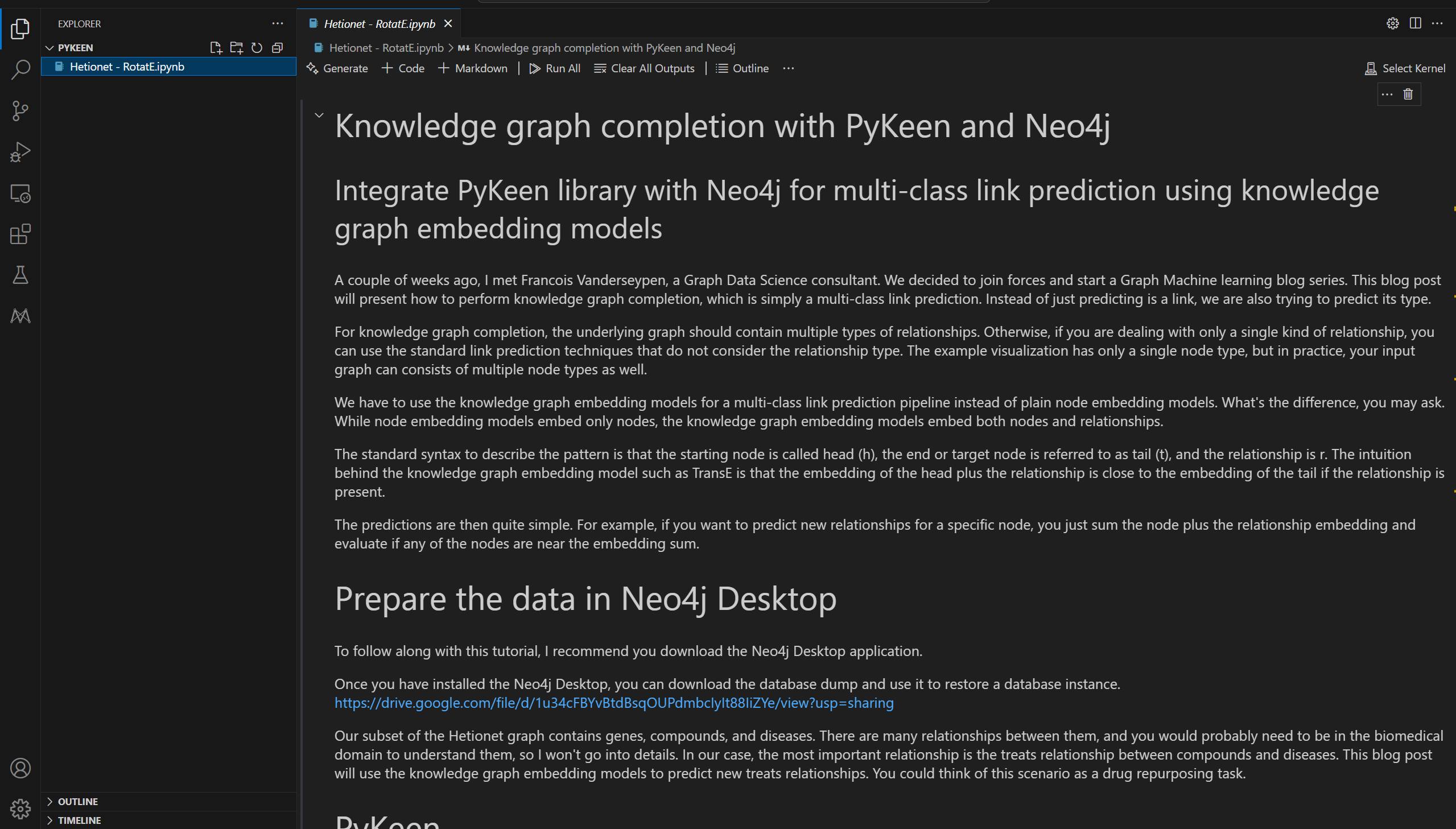
若右键菜单中无此选项，则可以打开一个新的命令行，进入所在目录。



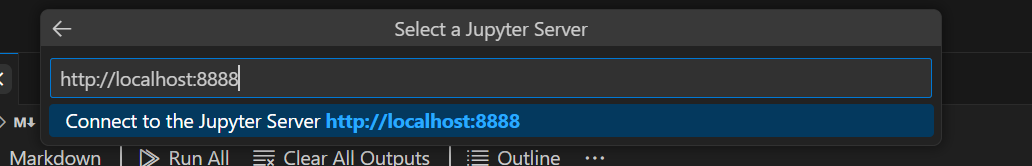
记录token（地址中token=之后的部分，如本案例中token前6位为50d50a）



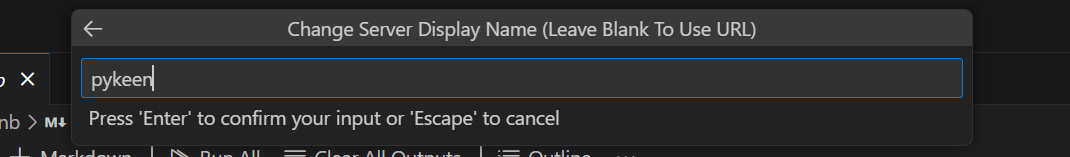
1. 在vscode 中打开pykeen-neo4j.ipynb文件。



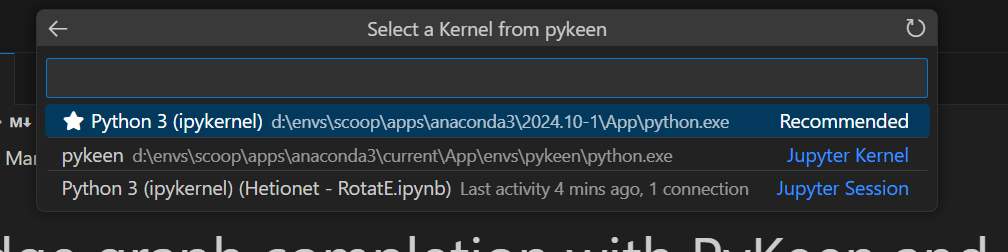
1. 在右上角点击select kernel -> Existing Jupyter Server，输入server地址。



1. 将（1）中记录的token作为password输入。
2. 起名为pykeen。



1. 选择pykeen。



1. 实验原理与步骤
2. 实验目的

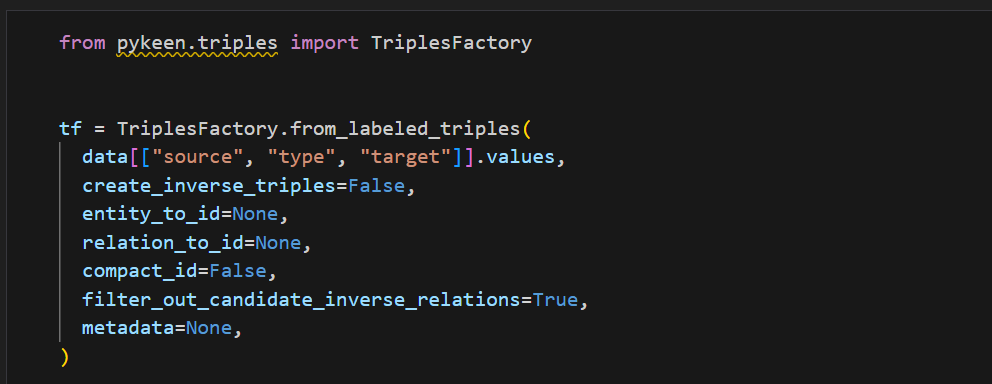
本实验将利用Hetionet数据集，实现图谱的链接预测。

1. 数据集介绍

Hetionet数据集是一个生物信息学的知识图谱，总计包括47031个节点和2250197条边，本实验使用其中的子集。

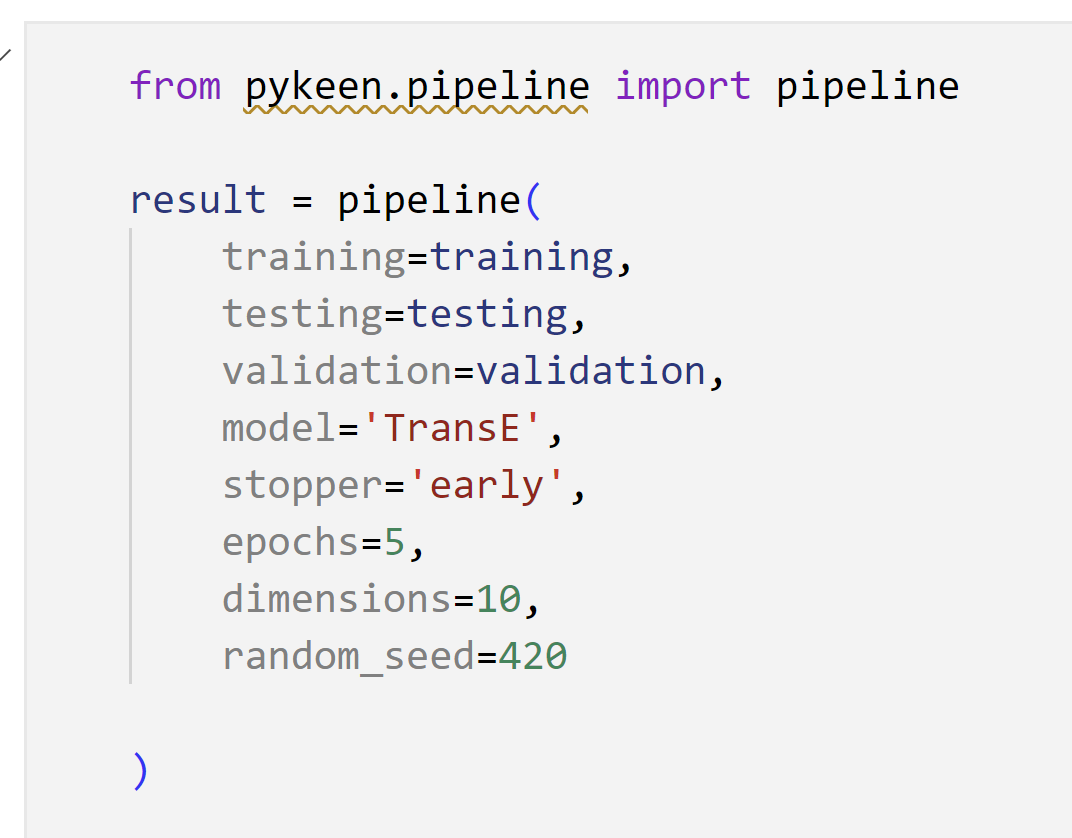
1. 构建三元组

Pykeen可以根据图谱的三元组信息，构建出知识图谱的内部表示。

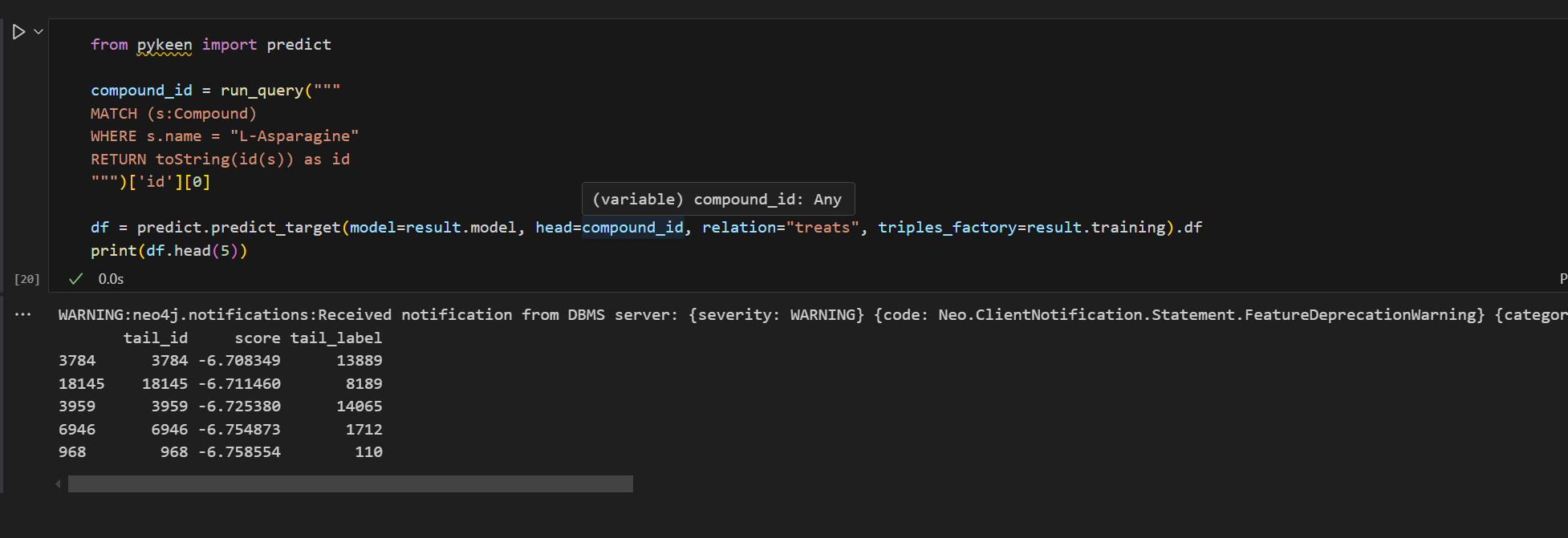


1. 训练知识图谱嵌入模型

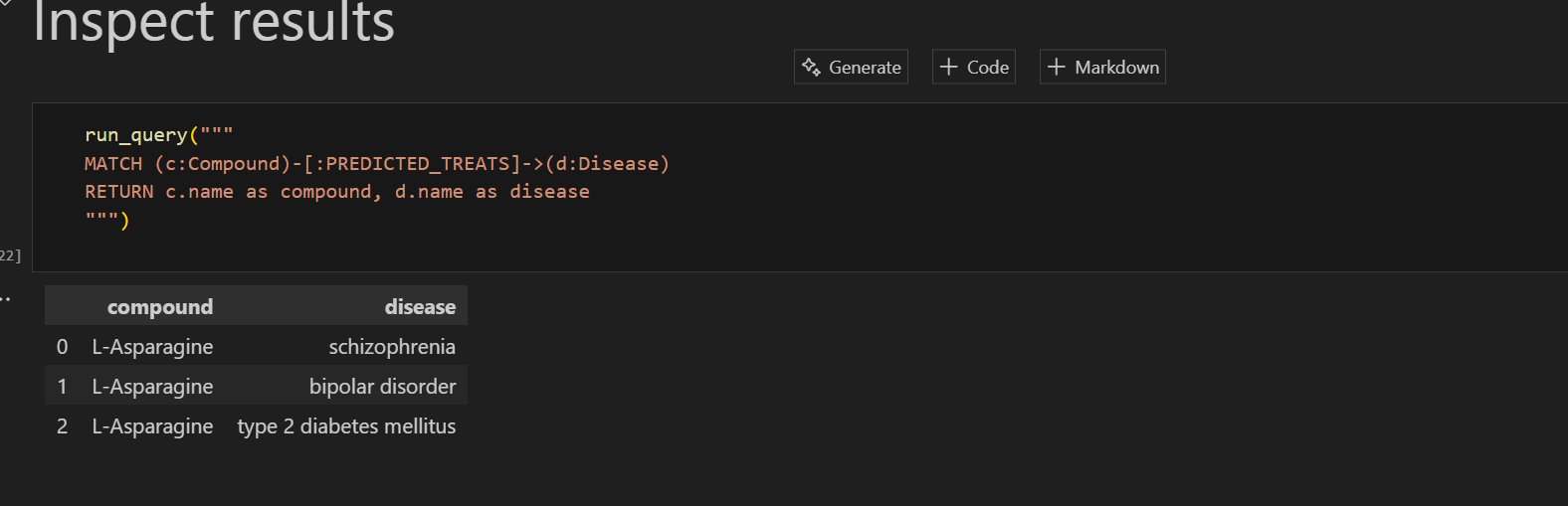
Pykeen提供了pipeline方法，用于快捷训练知识图谱嵌入模型，本次实验选择使用TransE算法。



1. 执行多路链接预测。



1. 存储回neo4j。
2. 查看预测结果



1. 执行notebook代码

【实验任务】完成notebook文件pykeen-neo4j.ipynb中的内容：

将notebook每个代码段执行成功的输出截图粘贴在下面

【代码段1执行截图】：

【代码段2执行截图】：

……