机器学习Agent大作业附加题

背景

随着人工智能和机器学习技术的快速发展，越来越多的实际问题可以通过构建有效的机器学习模型加以解决。然而，构建一个高性能的机器学习系统往往需要专业的知识和大量的人工干预，包括数据预处理、特征工程、模型选择、超参数调优等多个复杂环节。这不仅提高了应用门槛，也限制了机器学习技术在更广泛领域的普及。

LLM-Agents能自主规划和执行任务，不仅能调用外部工具完成真实操作，还能具备记忆和协作能力。相比普通大模型，Agent 更智能、更高效，能持续执行复杂任务，实现从“语言理解”到“实际行动”的跨越，非常适合处理机器学习这样复杂耗时的任务。

在本次大作业附加题中，同学们需要构建属于自己的ML-Agents System(Machine Learning LLM Agents System)，并通过真实的机器学习比赛来测试系统的性能。

本次附加题为开放性大作业，旨在鼓励**学有余力**的同学们进行**自主探索与实践**。得分将以**额外加分**的形式记入成绩。请同学们根据自身时间与能力情况，尽力完成相关内容，无需强求全部实现或获得最终完整结果，不以是否全部完成作为唯一评判标准。

MLE-Bench-tiny

简介

MLE-Bench是Open AI在2024年发布的一个benchmark，由75个从kaggle网站精心挑选的机器学习比赛组成，旨在评价智能体处理自动化机器学习这一复杂、慢反馈任务的能力，论文入选ICLR 2025 Oral，这一benchmark受到了学界和工业界的广泛关注。

完整的MLE-Bench数据集大小约3TB，考虑到计算资源的限制，在本次大作业中，我们仅从MLE-Bench中选择几个小型比赛组成MLE-Bench tiny进行测试，具体比赛名称如下

代码块

```
1   nomad2018-predict-transparent-conductors #demo示例比赛
```

```
spooky-author-identification
dog-breed-identification
```

对应的kaggle比赛官方链接为

代码块

```
https://www.kaggle.com/competitions/nomad2018-predict-transparent-conductors
https://www.kaggle.com/competitions/spooky-author-identification/overview
https://www.kaggle.com/competitions/dog-breed-identification/data
```

数据集下载与使用说明

为方便同学们下载，这三个比赛的MLE-Bench版数据集已经上传到交大云盘（链接：https://pan.sjtu.edu.cn/web/share/2432233eeb2a83397e65bb8f9ce89ef1），每个比赛解压后数据格式如下

代码块

```
./nomad2018-predict-transparent-conductors/
└── prepared/
    ├── private/
    │   └── test.csv
    ├── public/
    │   ├── test/
    │   ├── train/
    │   ├── description.md
    │   ├── sample_submission.csv
    │   ├── test.csv
    │   └── train.csv
    ├── baseline.json
    ├── grade.py
```

其中：

- private文件夹存放到是最终提交的预测文件的真实值，用于最终评价打分，不可泄漏。
- public文件夹存放训练需要用到的数据，其中：
  - description.md包含详细的比赛任务描述
  - sample_submission.csv是一个最终要提交的预测文件的示例（不包含实际真实值），主要用于说明提交文件的格式
  - train.csv和train文件夹包含训练集数据以及对应的label或者ground truth
  - test.csv和test文件夹包含测试集数据（不含label或ground truth）

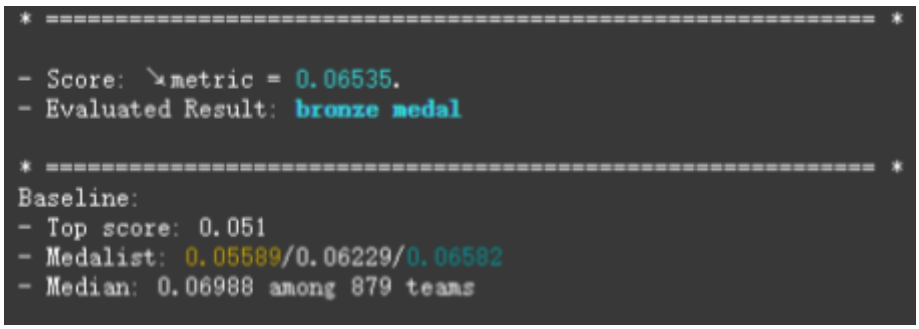
- baseline.json记录了真人在比赛上完成之后得到的奖牌线
- grade.py是一个评测函数，可用于对提交的预测文件进行打分评价

在MLE-Bench上进行评测，agent需要根据public文件夹下的数据和任务信息（剩余文件仅在评测最终的提交文件时使用，对agent不可见），自主在训练集上完成模型的训练，并使用训练得到的模型在测试集上进行预测，生成一个预测文件submission.csv。

生成submission.csv后，手动运行grade.py，即可得到一个最终分数和等第。使用方式为：

```
代码块
1   python grade.py -s "your submission file path" -g "your ground truth file path"
```

如果submission.csv格式正确，输出如下



metric前方的箭头表示这个任务的分数是越高越好或越低越好，等第从高到低依次为为金牌/银牌/铜牌/超过半数/未超过半数（但格式合法）/提交格式不合法，下方baseline也提示了对应等第（金牌/银牌/铜牌）的分数线。

计算资源

API

modelscope为普通开发者提供了每日2000次的模型调用额度，可以使用各种开源模型，在构建LLM-Agents时可以使用。具体使用方法如下。

1. 登录modelscope官网，网址为https://modelscope.cn/my/overview
2. 根据https://modelscope.cn/docs/accounts/aliyun-binding-and-authorization中1.1,1.2,1.3部分的指引完成阿里云账号的绑定，主账号授权和主账号开通&使用云服务
3. 根据https://modelscope.cn/docs/model-service/API-Inference/intro的指引，在安装完成openai库后，用自己的api_key尝试调用模型，示例调用程序为

```
代码块
1   from openai import OpenAI
```

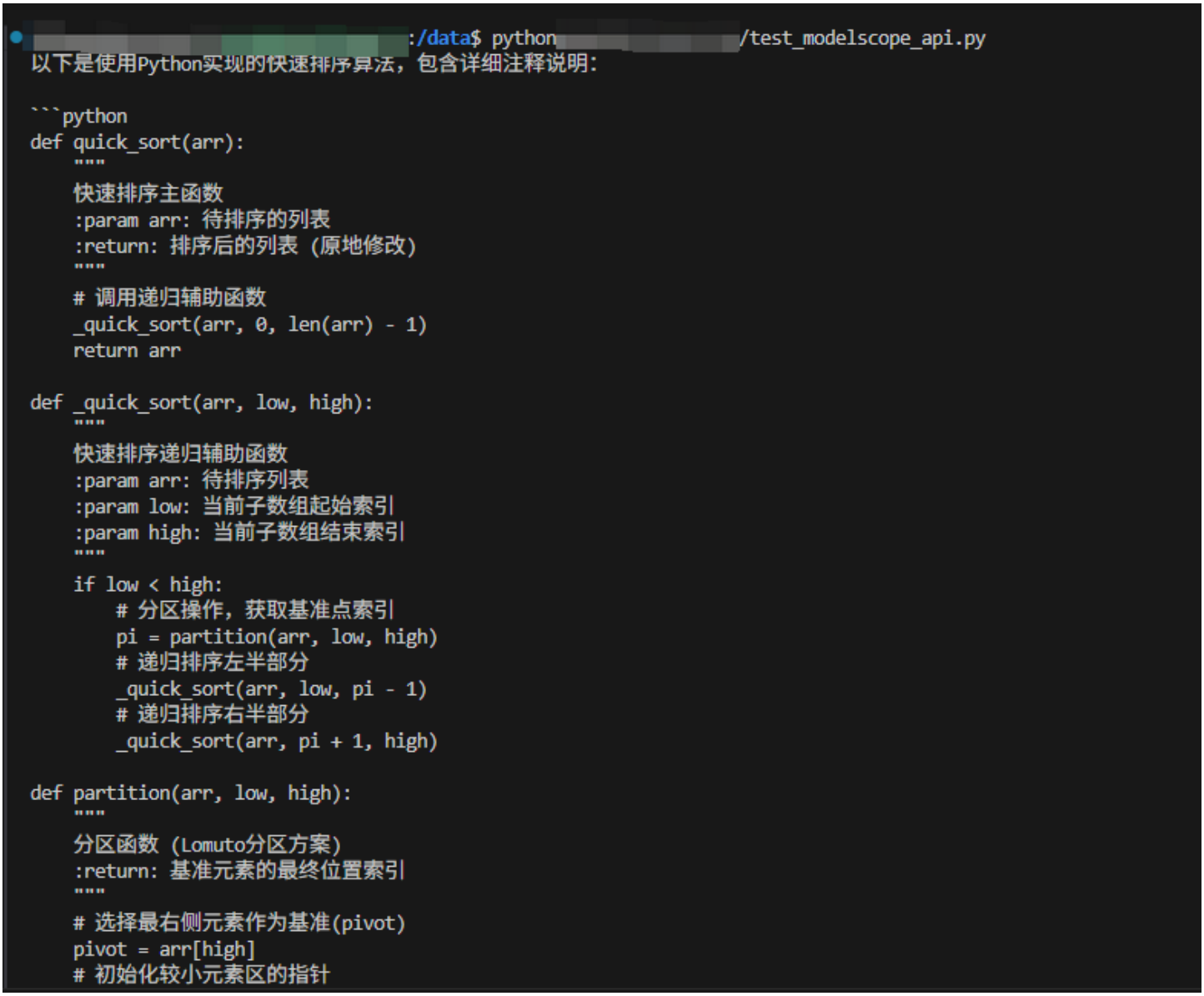
```python

client = OpenAI(
    api_key="MODELSCOPE_ACCESS_TOKEN", # 请替换成您的ModelScope Access Token
    base_url="https://api-inference.modelscope.cn/v1/"
)

response = client.chat.completions.create(
    model="Qwen/Qwen3-235B-A22B-Instruct-2507", # ModelScope Model-Id
    messages=[
        {
            'role': 'system',
            'content': 'You are a helpful assistant.'
        },
        {
            'role': 'user',
            'content': '用python写一下快排'
        }
    ],
    stream=True
)

for chunk in response:
    print(chunk.choices[0].delta.content, end='', flush=True)
```

如果调用成功，会在终端看到流式输出：



作业要求

综合考虑难度、工作量等因素，以下两项只需任选其一完成即可。

改进现有的ML-Agent框架

为方便同学们理解任务内容和上手，我们提供了一个可以直接运行的现有的ML-Agent框架ML-Master（https://github.com/sjtu-sai-agents/ML-Master），且后续（约11月初）会有一份详细教程帮助大家上手和熟悉此框架，跑通demo。更多内容请参考后续发布的文档"ML-Master使用教程"。

选择此项的同学可以在demo跑通的基础上进一步的开发和改进，提升其性能，包括但不限于提示词工程、算法优化、性能优化等。

构建属于自己的ML-Agent框架

根据自己的想法开发属于自己的ML-Agent框架，此项没有额外要求，但请确保你的框架能适配MLE-Bench的数据集格式（详见数据集下载和使用说明一节），以完成MLE-Bench相关任务的评测。更多

相关信息请参考https://arxiv.org/abs/2410.07095以及https://github.com/openai/mle-bench。

在评测时，请仍然参考和遵守文档"数据集下载和使用说明"一节中的评测流程。

作业提交内容

代码提交

将所有代码、日志文件以及说明文件打包成zip文件并提交到canvas

报告提交

将大作业报告以pdf格式提交至canvas，报告格式、字数不限，但应做到简洁、清晰，可以描述完成的内容，取得的结果，也可以描述自己遇到了什么问题，如何尝试解决等。以下内容如果完成，推荐大家在报告中明确展示：

- ml-agent框架能成功运行
- ml-agent框架能在运行一段时间后按要求生成submission.csv文件
- 生成的submission.csv文件格式合法，grade.py在终端打印了具体分数
- 生成的submission.csv文件得分较好，grade.py在终端输出了获得奖牌

本项作业作为附加题更侧重锻炼大家的工程能力和解决问题的能力，不推荐大家卷报告字数。

评分

本次附加题为开放性大作业，旨在鼓励**学有余力**的同学们进行自主探索与实践。请同学们根据自身时间与能力情况，尽力完成相关内容，无需强求全部实现或获得最终完整结果，不以是否全部完成作为唯一评判标准。

在评分方面，将根据**实际完成情况**进行**综合评定**，包括但不限于实现程度、思路设计、最终结果、尝试过程及创新性等因素。

注意：禁止包括但不限于通过泄露测试集真实预测值、人为修改grade.py代码等方式来获取更好的分数，一经发现，后果自负。

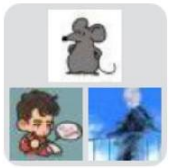
参考文献

Title	url
MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering	https://arxiv.org/abs/2410.07095
	https://arxiv.org/abs/2310.03302

MLAgentBench: Evaluating Language Agents on Machine Learning Experimentation	
AIDE: AI-Driven Exploration in the Space of Code	https://arxiv.org/abs/2502.13138
ML-Master: Towards AI-for-AI via Integration of Exploration and Reasoning	https://arxiv.org/abs/2506.16499
R&D-Agent: An LLM-Agent Framework Towards Autonomous Data Science	https://arxiv.org/abs/2505.14738
ML-Agent: Reinforcing LLM Agents for Autonomous Machine Learning Engineering	https://arxiv.org/abs/2505.23723

答疑群

对大作业附加题有任何问题，扫码加入答疑群。



群聊：2025 Fall 机器学习进阶项目

该二维码7天内(11月2日前)有效，重新进入将更新