



Datawhale AI夏令营 Task4

Task4：项目优化与进阶

💡 也许你是第一次参加类似的创新应用开发赛，也许你是个中好手，但我相信你在类似的赛事过程中，总会遇到一些困难和心酸，在这里我们想给你一些简单的建议，也很欢迎你来一起共创！

直播录制：

https://meeting.tencent.com/v2/cloud-record/share?id=d82bb3a5-bf89-48c3-96a8-12a3b93e9cd2&from=3&is-single=false&record_type=2

如何更好地组队？让项目尽量能够完成？

在黑客松比赛这种时间非常紧张的项目开发中，组建一个高效协同的团队至关重要。

1. **尽量找到正确的人**：最好技术栈互补
 - a. 一般至少：前端(最简单的就是Gradio/Streamlit)、后端(偏大模型选型与部署)、产品设计都要有人
 - b. 选队友的时候最好设置点门槛+语音聊半小时，才能保证靠谱点
 - c. **队长需要能扛事，对目标相对清晰，能带领团队往前跑，能积极争取资源支持**
 - d. 队伍里至少需要会技术的人，队长尽量要有一些技术视角，没有的话找搭子互相聊出来，聊清楚
2. **尽量让所有人都能发挥所长**：最好是让大家做擅长的事，
 - a. 首先，明确团队成员的技能和兴趣点，尽量确保每个人都能发挥所长，以及学到/实践到自己感兴趣的，并顺利完成。
3. **设定清晰的目标和期望**：队长需要把控风险
 - a. 要明确赛事核心的要求是什么，盘点目前的能力和资源能做到什么样子
 - b. 让每个人都明白团队的方向和自己的职责，并尽量预估合理的时间规划。
4. **集思广益**：开放沟通、互相促进和激发创意
 - a. 保持开放的沟通渠道，鼓励团队成员提出想法和反馈，有助于团队创新和问题解决

如何跟队友做好沟通？

有效的沟通建立在尊重和理解的基础上。

- 定期举行团队会议，分享进度和挑战，确保信息透明。
- 使用项目管理工具(最简单粗暴的就是TODOList)来跟踪任务分配和完成情况。
- 最重要的是，培养一种积极的团队文化，鼓励团队成员之间的相互支持和鼓励。
- 多在群里互相分享，互相点赞~

如何设计产品的方向？ 哪些人应该参与进来？

产品方向应基于市场需求和团队优势。进行市场调研，了解潜在用户的需求和竞争对手的现状。分工时，考虑每个成员的专业背景和个人偏好，确保团队成员能在自己擅长的领域发光发热。

如何迭代项目？——ChatTests有想过这些问题

ChatTests是一个专注于教育领域的大模型应用，旨在通过AI技术提升学习体验。它不仅考虑了上述团队建设和项目管理的问题，还深入探讨了产品功能和用户体验。

ChatTests目前的功能和体验方式

https://modelscope.cn/studios/chattests/chattests_analysis

1. **AI出题**：利用AI生成题目，覆盖多种知识点。
2. **题目效果体验**：用户可以实时体验AI生成题目的质量。
3. **题库管理**：用户可以保存和管理生成的题目。
4. 衍生：**顺序刷题模式**（待做）：计划添加按顺序刷题的功能。
5. 衍生：**十万个ChatTests**（待做）：探索更多个性化的学习路径，该模式是让用户根据给定的图片/文字，进行提问，然后优化题干后生成相应的单选题 & AI给问题进行评分

ChatTests目前的代码详解

1. 选型

Appbuilder技术栈

- Appbuilder网页使用
- Appbuilder相关组件能力：如OCR（光学字符识别）

星火技术栈

- 星火助手
- dwspark、sparkai

部署使用免费CPU资源：魔搭、AI Studio部署

核心思路：先堆免费有资源的能力、再做更多稳定性和UI的考虑

2. 整体架构

最近在学习modelScope_studio代码，进行文件管理

目前拆分为ChatTests整个功能逻辑的文件夹

3. 模块详解

1. 知识文本识别
2. AI 出题
3. 题目展示
4. 题库管理

- 5. 刷题功能
- 6. 衍生能力

ChatTests的设计理念和开发故事

1. 梦开始的地方

学习心得：给组员和一起Hackathon过的同期伙伴

来自天南海北的我们，因为DataWhale这场Hackathon的缘分，聚在一起，有机会共同努力把 **一个想象落地**。

我们所有人都携带着自己的**坚持、真心和热情**，满怀激情地相遇。从一开始的破冰、到拆分任务到个人各自努力，再到一点一点把技术方案确定、根据实际情况改良，最终使用合适的框架把一个具体的demo搭建出来，有了具体的目标不断优化，回忆起来真是感触良多。

这一路有些坎坷，有些意外，但好在一切都妥善地被我们处理好，即使技术能力遇到滑铁卢，我们还是顽强地找到了替代的方案，在这里真的**由衷地感谢积极付出的你们**。


有些努力，虽然在目前阶段性的成果中没有很好地对外展现，但所有的努力累积，才让我们走到这里，没有白走的路！一切都有意义！

有伙伴可以依赖，一起进步，一起做事，互相支撑着前行，并做出了属于我们自己的作品，这一切真的很有意义！**我们真的很了不起！**

感谢你们带我走过一次次焦虑和放弃的边缘，我从你们身上，从我们这次获奖中明白了：要更平常心地看待一切成功和努力，不能随便妄自菲薄，做好眼前具体的事情，持续进步，持续提供优秀的想法和努力落地实现，就是在创造价值！

我们每个人都拥有奇思妙想，选择一个确定的道路一起前行，就已足够！**我们继续前行吧！**前路漫漫，有缘继续前行也好，各自顶峰相见也行，**不要忘记这次的能量，不要忘记我们彼此曾互相给过的支持！**

祝DataWhale越来越好！我们越来越好！！



Avalon

我们在未知的领域中学习了很多，很有趣。参加活动之前，我对ai应用抱有一腔热情，以及ai+教育的想法，但是缺乏python基础以及ai相关知识体系，苦于无法实现自己的想法。加入小队以后，发现好多志同道合和充满热情的小伙伴，看到datawhale的课程帮助，感觉自己的靠谱的小伙伴在这条路越走越远了！大家太棒了！ 特别感谢cap！

在本次大语言模型应用的学习中，我学到了很多，团队内的各位大佬分工明确，代码能力极强，看到了他们对专业知识的热爱。把AI模型部署到自己搭建的网站中，将AI变成一个自己学习过程中的学习助手。不过遗憾的是自己不太擅长前后端相关工作，这次比赛有些拖团队的后腿，完完全是沾团队里其他大佬的光，一些很简单的数据处理任务都没有做好，在这里向团队里其他成员致歉！！如果以后有机会参加其他的AI比赛，希望可以请各位大佬一起，再接再厉，努力向优秀的大佬学习！

- 📄 🍷 ChatTests项目详细说明文档
- 📄 ChatTests-DataWhale优秀作品答辩分享.pdf
- 📄 大模型启发演讲：ChatTests落地之路 - Amy.pdf

2. 开发迭代经验

- 以下偏创意赛方向的经验思考，乱拳打死老师傅
- 1. **快速写前端界面效果**：通过快速原型展示功能目标，促进团队理解和用户反馈。
 - 2. **代码缪斯源泉**：善于利用官方文档和社区资源，激发灵感。
 - 3. **从自己的场景出发**：基于个人经验和用户需求，进行功能设计。
 - 4. **遇到问题，思考解决方案**：灵活应对挑战，适时调整策略。
 - 5. **创意优先**：想到更多的创意点和对应的功能，进行组合设计
 - 6. **用户体验优先**：关注用户界面的清晰度和交互性。
 - 7. **功能更新与代码复用**：优化代码结构，实现多功能复用。
 - 8. **独立功能集成**：先独立实现功能，再集成到整体项目中。
 - 9. **把演示材料做好(如PPT)**：这个是不管是赛事还是项目发展都很需要的点，既是与团队对齐目标和远期愿景，又是 赛事评审/外部资源 最高效了解项目的方案，是替你说话的部分(很多时候这个材料不够好到不了答辩的环节)

更多内容可以参考：📄 4. 【代码级】AI应用开发入门、📄 AgentScope应用开发入门

3. 其他问题（征集 or 想说的）

1. idea从概念到落地的挑战：从概念到现实，克服技术和资源障碍

需要迅速掌握尚未熟练的新技术，这不仅考验着团队的学习能力，也考验着项目管理者决策与协调能力。当技术能力未能匹配创意时，需要寻找新的技术方案，通常可以参考现有项目或在 **学习群/作品交流群/应用开发发烧友群** 向领域专家求助，以找到解决问题的方法。

- 技术跟进：有时候想法超前于现有的技术积累，需要寻找新的技术点或方案来实现功能。
- 实践方法：可以通过观察和学习其他项目是如何实现类似功能的，借鉴代码实现方式。

将创意转化为实际产品往往伴随着多重挑战。

- 首先，创意的可行性需要经过技术评估，确保现有技术能够支撑设想的功能。
- 其次，团队需要具备跨学科的知识，包括但不限于技术、设计、市场和业务策略，以全面推动项目进展。
- 此外，资源限制、时间管理及市场动态也可能成为阻碍因素。

因此，团队需要制定详细的计划，设定优先级，并灵活调整策略以应对不确定性。

2. 如何保证生产场景中产品的稳定性

通常情况下，我们并不太需要考虑生产场景下的稳定性，如果遇到bug可以在交流群里提问，通过集体智慧解决技术难题。

对于比赛来说，在最初就考虑系统的可扩展性和健壮性，并不现实，一是这些是架构师能力的人才能很好地思考到，且需要大量实践；二是比赛的开发时间有限，如果过多考虑这些，时间成本会大大被内耗掉，而无法做出更多更好更实用的功能

至于真的需要产品化、商业化的时候，需要扩展支持并发使用和投入大规模场景时，可以再去考虑相关的问题，就找专家咨询，甚至付费咨询，一般的操作可能涉及：

- 采用微服务架构、冗余设计和自动化测试可以帮助提高系统的可靠性。
- 同时，持续集成/持续部署（CI/CD）流程可以确保软件更新的平滑过渡，减少生产环境中的故障风险。
- 监控和日志记录机制则能及时发现问题并迅速响应，从而保障用户体验。

3. 开源代码和idea被借鉴和学习会怎么想？

Amy自己觉得，从他人项目中汲取灵感并将其融入自身产品，是推动产品迭代与进步的有效方式。每个创意都是独特的，通过相互启发和学习，可以促进整体生态的繁荣。当然，平衡版权与创新是开源文化中的重要议题，适时逐步开源核心内容是一种常见的策略。

开源精神的核心在于共享和协作。大多数开源贡献者欢迎他人学习和借鉴自己的工作，因为这促进了技术的传播和创新。

不过尊重原作者的版权和贡献同样重要，这意味着在使用开源代码时应遵守相应的许可协议，给予适当的署名和归因。

同时，鼓励回馈社区，无论是通过修复bug、添加新功能还是分享自己的改进，都是对开源生态的积极贡献。

4. 如何设计产品，让产品更有趣？需求是怎么来的？

观察队友及个人的信息关注点来提炼产品需求，解决团队的痛点，做大家会用的应用

团队成员的关注点和痛点往往是产品创意的源泉。此外，市场需求调研也不容忽视，产品设计应融合用户反馈与市场趋势，以确保产品既实用又具有吸引力。在设计细节方面，例如题目的难度量化、跨领域个性化处理、题型扩展等方面，需要综合考虑知识体系与用户能力水平，以实现系统的智能化出题。

- 需求来源：关注自己和队友的信息需求，解决痛点，作为产品设计的核心。
- 市场导向：部分人更注重市场需求，产品设计细节需逐步完善。

- 创意征集：通过社群收集创意(让大家体验和提需求共创)，用以改进产品，增加趣味性。

设计有趣的产品通常还涉及到理解目标用户的需求和兴趣。可以：

- 通过引入互动元素、游戏化机制或个性化体验，可以提升用户参与度。例如，设置成就系统、排行榜或虚拟奖励可以激励用户持续使用产品。
- 同时，保持界面简洁直观，避免过度复杂化，也能让用户感到轻松愉悦。
- 创意和幽默感的融入，如有趣的动画、俏皮的文案或独特的视觉风格，也能让产品脱颖而出。

5. 产品设计细节：题目的难度如何衡量？不同领域的出题如何做个性化处理？题型有考虑扩展吗？如何根据一大堆的知识进行系统性出题？

衡量题目难度可以基于几个维度：知识深度、所需理解层次、解题所需时间以及用户反馈。通过算法分析，可以建立一个动态难度调整系统，确保题目既具有挑战性又不至于令人沮丧。个性化处理涉及数据分析和机器学习，根据用户的历史表现和偏好定制题目。题型的扩展应该考虑多样性和实用性，从选择题到开放问答，再到实践操作，以适应不同学习风格和目标。系统性出题则需构建知识图谱，确保题目覆盖广泛且均衡，避免偏颇。

6. AI有没有生成过一些让你印象深刻的东西？



AI的奇思妙想展示了其独特优势，但如何在准确性和创造性之间取得平衡，仍是待解之谜。这不仅关乎技术，也涉及到对AI应用边界和潜力的探索。

7. 队友从何而来？队友因何而走？

队友的加入源于共同的兴趣和愿景，他们的离开有时则显得无迹可寻。

在DataWhale社区，我见证过很多队友的成长与变迁，团队成员追求个人项目或研究方向的转变是正常现象，重要的是能够持续在社区中相遇，并为彼此的进步感到高兴。

8. 项目能发展的核心原因是什么？对负责人的要求有哪些？

项目持续成长的核心在于负责人对项目的坚持、持续思考与创新。能有相似想法的人数不胜数，但相信并坚持做下去和迭代的人并不多，坚持做难而正确的事情并不容易，但这是项目能长期发展的关键。

能有人不断思考项目针对的问题场景的解决方案是项目前进的关键，这不仅能促进现有问题的解决，还能催生新的创意与优化方案。负责人应具备长期视野，引领项目向着更大、更好的方向发展。

往往项目的负责人是对项目的感情最深、最想把项目往前推进的人，其他队友可能会来来去去，互相有各自的坚持的项目，然后互相扶持，也挺好的。

- **队长/负责人** 需要能扛事
 - **目标明确**：清晰地知道赛事的重要目标是什么，分清主次，良好地规划项目周期
 - **任务分解**：将大型项目拆解为多个小任务，分配给具备相应技能的成员，确保每个人都能贡献自己的优势。
 - **学习与培训**：提供必要的学习资源，帮助团队成员提升技能，特别是对于新技术的学习。
 - **沟通机制**：建立有效的沟通渠道，确保信息流通顺畅，团队成员能够及时解决问题和分享进展。

- **PPT很重要**
 - 精心设计产品介绍和演示，确保评委能够全面理解产品的价值和创新点。
 - 需要更加注重项目与赛事需求的契合度，明确展示项目亮点，确保评审能够准确理解项目价值。


此外，PPT的视觉呈现与内容组织同样重要，良好的美学布局和清晰的逻辑结构能够增强项目吸引力，从而在赛事中脱颖而出。




- **在有限的时间内持续优化项目**
 - **用户体验**：进一步提升用户界面的友好性和操作流畅性，使产品更加易于使用。
 - **功能扩展**：根据竞赛要求和用户反馈，适时增加新功能，以增强产品的竞争力。

9. ChatTests要往哪儿走？

ChatTests会继续为大家带来更多好的开发思路和努力实现最初设计时想完成的样子，不断推进，也不断陪着大家一起成长，后续的应用开发学习活动大家会很频繁地看见ChatTests，ChatTests也会努力不断给大家带来更多更新更有价值的功能和模型能力，努力跟上AI的发展，陪大家一起走过更多项目的进阶之旅~

如何针对赛事需求更好地阐述项目？

有需要的话，私聊  周理璇 (Amy) 获取 ChatTests 的项目介绍PDF吧~

 本教程主要由  周理璇 、  徐炽明 写作贡献，经过多位助教内测和建议修改，期待你的点赞和评论~