> 在开工之前我们需要先花一个短周期搜集数据&做准备工作。

项目准备工作总结

一、数据准备 (最核心)

高光谱数据是项目的基石，没有合适的数据，系统就无法运行和验证。

1. **寻找合适的高光谱数据集：**
    - **要求：** 包含海岸带区域（滩涂、盐沼、水体、建筑等），最好提供**真实地物分类标注（Ground Truth）**，这样才能进行模型训练和准确性评估。
    - **数据特点：** 高光谱数据通常较大，处理起来需要一定的计算资源。
    - **替代方案：** 如果找不到完全符合的海岸带高光谱数据，可以考虑：
        - **其他地物分类的高光谱数据：** 先用其他典型地物数据（如农作物、森林）进行系统框架的搭建和功能实现，后续再替换为海岸带数据。
        - **合成数据或模拟数据：** 对于原型系统，可以考虑使用工具生成一些具有高光谱特征的模拟数据，以验证分类流程。
2. **获取高光谱数据处理工具/库：**
    - 处理高光谱数据需要专门的库，用于读取、操作、可视化多维数据。

相关链接：

- **高光谱数据集：**
    - **Pavia University/Center Data:** 经典的城市高光谱数据集，虽然不是海岸带，但可以用于学习和验证分类算法。
        - http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
    - **Indian Pines Data:** 另一个经典农田数据集。
        - http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
    - **IEEE GRSS Data Fusion Contest:** 每年都会发布新的遥感数据集，有时会包含高光谱数据和标注，可以关注往年数据。
        - https://www.grss-ieee.org/technical-committees/image-analysis-and-data-fusion/data-fusion-contest/
    - **Earth Engine Data Catalog (需要一定的编程能力):** 包含了大量卫星遥感数据，其中一些可能是高光谱或多光谱数据，可以尝试筛选海岸带区域。
        - https://developers.google.com/earth-engine/datasets/catalog
    - **特定科研机构或大学：** 搜索相关研究论文中引用的数据集，有时可以联系作者获取。
- **高光谱数据处理库 (Python):**
    - **spectral:** 专门用于高光谱遥感数据处理的Python库。
        - GitHub: https://github.com/spectralpython/spectral
        - 文档: http://www.spectralpython.net/
    - **scikit-image / OpenCV:** 用于一般的图像处理操作，如降噪、可视化。
        - scikit-image: https://scikit-image.org/
        - OpenCV: https://opencv.org/
    - **gdal:** 用于读取和写入各种遥感数据格式（如 GeoTIFF、ENVI格式）。
        - https://gdal.org/

二、技术选型与模板 (前后端分离)

本项目涉及前后端，需要对技术栈进行选择。

1. **前端技术栈选择：**
    - 根据个人或团队熟悉度选择 **React / Vue / Angular** 中的一种。如果追求快速原型，也可以考虑纯 **HTML/CSS/JS + AdminLTE**。
    - **UI 框架/组件库：** 搭配主流的 UI 库，如 Ant Design (React), Element UI (Vue), Material-UI (React), Bootstrap (通用)。
    - **图表库：** 用于光谱曲线、混淆矩阵、性能指标可视化，如 ECharts, Chart.js, Plotly。
    - **地图可视化 (可选，但会增强):** 如果需要显示地理位置上的分类结果，可能需要集成地图库，如 Leaflet 或 OpenLayers。
2. **后端技术栈选择：**
    - **编程语言：** **Python** 是首选，因为它在高光谱数据处理和机器学习/深度学习领域有最成熟的生态系统。
    - **Web 框架：** **Flask** (轻量级，适合API服务) 或 **Django** (功能更全面，适合大型应用)。
    - **AI/ML 库：** **scikit-learn** (传统机器学习模型，如SVM、随机森林), **PyTorch / TensorFlow / Keras** (深度学习模型，如CNN)。
    - **数据处理：** **NumPy, SciPy, Pandas** 用于高效的数据操作。
3. **前后端通信协议：** RESTful API 是主流选择。

相关链接：

- **前端模板 (已在前一轮提供，这里再强调):**
    - **React:** Ant Design Pro, Material Dashboard React
    - **Vue:** Vue Element Admin, Material Dashboard Vue
    - **HTML/CSS/JS:** AdminLTE
- **图表库：**
    - **ECharts:** https://echarts.apache.org/zh/index.html
    - **Chart.js:** https://www.chartjs.org/
    - **Plotly:** https://plotly.com/ (也常用于 Python 后端生成交互式图表)
- **后端框架 (Python):**
    - **Flask:** https://flask.palletsprojects.com/
    - **Django:** https://www.djangoproject.com/
- **AI/ML 库 (Python):**
    - **scikit-learn:** https://scikit-learn.org/
    - **PyTorch:** https://pytorch.org/
    - **TensorFlow:** https://www.tensorflow.org/
- **地图库 (可选):**
    - **Leaflet:** https://leafletjs.com/
    - **OpenLayers:** https://openlayers.org/

三、开发环境配置

在开始编码前，需要搭建好开发环境。

1. **代码编辑器：** VS Code (推荐，支持多种语言和插件)。
2. **Python 环境：** Miniconda/Anaconda (推荐，便于管理虚拟环境和科学计算库)。
3. **Node.js / npm (或 yarn)：** 如果选择 React/Vue/Angular 前端框架，用于包管理和构建。
4. **版本控制：** Git (配合 GitHub/GitLab 等平台进行项目管理和协作)。

相关链接：

- **VS Code:** https://code.visualstudio.com/
- **Miniconda:** https://docs.conda.io/en/latest/miniconda.html
- **Node.js:** https://nodejs.org/
- **Git:** https://git-scm.com/

四、知识储备

项目涉及多个领域，需要补充相关知识。

1. **高光谱遥感基础知识：**
    - 高光谱数据的特点（高维度、连续光谱）。
    - 常见的预处理方法（降噪、波段选择、维度降低）。
    - 地物光谱特征。
2. **机器学习/深度学习基础：**
    - 分类算法原理 (SVM, Random Forest, CNN)。
    - 模型训练、验证、测试流程。
    - 性能评估指标 (准确率、Kappa、混淆矩阵)。
3. **Web 开发基础：**
    - 前端 (HTML/CSS/JavaScript, 选定的框架)。
    - 后端 (选定的框架，RESTful API 设计)。
    - 前后端交互原理。

相关链接：

- **高光谱遥感入门教程/书籍：**
    - 可以搜索“高光谱遥感原理与应用”、“Hyperspectral Remote Sensing”等关键词，寻找大学课程或在线教程。
    - **ENVI 软件教程：** 虽然是商业软件，但其教程中对高光谱数据处理的原理和步骤有很好的解释。
- **机器学习/深度学习在线课程：**
    - **Coursera / edX / Udacity：** 搜索 Andrew Ng 的机器学习课程、深度学习专项课程等。
    - **Fast.ai：** 侧重于实践的深度学习课程。
- **Web 开发教程：** 根据你选择的前后端框架，寻找相应的官方文档、教程和社区资源。

总结来说，项目的成功关键在于：

1. **高质量的数据集：** 尤其是带有准确标注的海岸带高光谱数据。
2. **合理的架构设计：** 如何组织前后端代码，如何设计 API 接口。
3. **模块化开发：** 将数据导入、预处理、模型、可视化等功能拆分成独立模块。
4. **迭代式开发：** 从最简单的功能开始，逐步完善和扩展。