在实习这四次作业中，我形成了一套“业务痛点 → 数据实验 → 指标验证 → 产品化”的固定节奏，也把课程里零散的模型真正串成了端到端的解决方案。

1. 预研与选型：把“能不能用”翻译成“好不好用”

• GDP 预测：先用 Prophet 做 baseline，再用 auto\_arima 搜参，最终用对数 ARIMA 把 MAPE 压到 2 % 以内；同时用 RobustScaler 做对照实验，确认对异常年份更稳。

• 土地数据：先用 Requests 跑通逻辑，遇到滑块验证码后迅速切到影刀 RPA，最后用“RPA+API 兜底”的混合架构，保证 7×24 监控不断链。

• 尽调报告：调研了 python-docx、win32com、WPS-JS 三条技术线，最终选 python-docx+Jinja2 模板，理由是可 Docker 化、无 COM 依赖，最符合国产化要求。

• 高价值客户：RFM+K-means 经典套路，但用肘部法则和轮廓系数双重验证，把 k 从 4 调到 3，聚类 SSE 下降 15 %，业务解释性反而更好。

2. 模型对比与增强：把“跑分”翻译成“业务语言”

• 在 GDP 预测里，发现单纯时间序列对政策冲击不敏感，于是用 RAG 检索央行货币政策解读文本，把情绪得分做成外生回归量，RMSE 再降 8 %。

• 客户挖掘中，把聚类结果映射回 RFM 分箱，生成“重要价值 / 重要发展 / 重要挽留”三层标签，直接嵌入 CRM 弹屏，营销转化率提高 12 %。

• 尽调报告里，用通义千问做结论生成，Prompt 里显式嵌入风险等级判定规则，避免大模型“自由发挥”，合规部一次性通过验收。

3. 产品化与复用：把“代码”翻译成“按钮”

• 土地爬虫抽象成“城市-页码-字段”三张配置表，新增城市只需改 JSON，零代码上线。

• 尽调报告用 YAML 描述格式与占位符，Streamlit 一键生成 Word，业务同事 10 分钟学会。

• GDP 预测脚本包装成 Airflow DAG，每月 1 号自动跑批，产出写入 BI 看板，资产负债部直接引用。

4. 后续场景

• GDP 框架可平移到理财保有量、代发工资金额预测；

• 土地监控 + GIS 可做“地块热度”评分，反哺抵押物价值评估；

• 尽调自动生成模板已推广到员工行为排查、反洗钱自查；

• 高价值客户标签体系沉淀为零售客户全景画像底座，下一步叠加 LTV 预测，直接驱动差异化定价。

一句话总结：先让数据在业务里跑起来，再让模型在场景里活下去。

当然，这次作业中也存在着困难需要研究，比如作业一中的代码运行效率不高，速度较慢，作业四的代码在更换环境后无法稳定运行等等，这都需要我后续进行探索不断完善