简历分析

一、简历文本预处理

1.1 多格式简历解析

简历格式可能存在以下几种，先进行解析：

- PDF
- DOC
- JPG/PNG

代码块

```
resume_parsers = {
    "pdf": {
        "技术方案": "PyPDF2 + pdfplumber + OCR备用",
        "输出要求": "保持原始格式结构，识别章节标题"
    },
    "doc/docx": {
        "技术方案": "python-docx + 格式解析器",
        "输出要求": "提取段落、列表、表格内容"
    },
    "图片格式": {
        "技术方案": "PaddleOCR/Tesseract + 版面分析",
        "输出要求": "识别文字+版面结构+表格"
    }
}
```

1.2 简历结构标准化

将提取出来的信息归类

代码块

```
standard_sections = {
    "个人信息": ["姓名", "性别","群众基础","联系方式", "邮箱", "求职意向"],
    "教育背景": ["学校", "学历", "专业", "时间", "GPA/排名"],
    "工作经历": ["公司", "职位", "时间", "工作描述", "业绩"],
    "项目经历": ["项目名称", "时间", "职责", "技术栈", "成果"],
    "专业技能": ["技术技能", "语言能力", "证书"],
    "自我评价": ["关键词", "职业目标"]
```

```
}
```

二、风险提示词

在分析简历的时候就要识别是否有造假风险，用于问题的生成

2.1 夸大风险检测

主要就是针对简历中是否出现程度夸大词、范围夸大词、成果夸大词、模糊责任词

代码块

```
exaggeration_indicators = {
    "程度夸大词": {
        "词汇列表": ["精通", "精通掌握", "深度掌握", "完全掌握", "极其熟练"],
        "风险等级": "高",
        "验证要求": "需要技术深度追问验证"
    },
    "范围夸大词": {
        "词汇列表": ["主导整个系统", "负责全部架构", "全面重构", "彻底改造"],
        "风险等级": "中",
        "验证要求": "检查职责与职级匹配度"
    },
    "成果夸大词": {
        "词汇列表": ["大幅提升", "显著改善", "极大优化", "革命性改进"],
        "风险等级": "高",
        "验证要求": "必须量化数据支撑"
    },
    "模糊责任词": {
        "词汇列表": ["参与", "协助", "配合", "了解", "熟悉"],
        "风险等级": "低",
        "验证要求": "澄清具体贡献度"
    }
}
```

2.2 数据真实性分析

主要针对简历中的业绩数据、时间逻辑和技术栈

代码块

```
data_authenticity_checks = {
    "业绩数据合理性": {
        "检查项": ["提升百分比", "用户数量", "系统性能指标"],
        "参考标准": "行业基准值 ±50%",
        "异常检测": "统计异常值检测"
```

```
        },
        "时间逻辑性": {
            "检查项": ["时间重叠", "时间断层", "工作时长"],
            "逻辑规则": [
                "教育和工作时间不能完全重叠",
                "项目时间应在工作时间内",
                "实习时间≤6个月"
            ]
        },
        "技术栈一致性": {
            "检查项": ["技术深度vs工作年限", "技术广度vs项目复杂度"],
            "评估模型": "基于经验曲线的技术掌握度模型"
        }
    }
```

三、能力评估提示词

3.1 技术能力标签提取

主要提取相关技术的专有名词作为关键词

代码块

```
    technical_capability_tags = {
        "编程语言": {
            "评估维度": ["掌握深度", "使用时长", "项目应用"],
            "评分规则": "基于上下文描述丰富度"
        },
        "框架工具": {
            "评估维度": ["熟练程度", "应用场景", "配置能力"],
            "评分规则": "是否有部署、优化经验"
        },
        "架构能力": {
            "关键词": ["架构设计", "系统设计", "技术选型", "性能优化"],
            "经验指标": "项目规模、并发量、数据量"
        },
        "软技能": {
            "沟通协作": ["团队协作", "跨部门沟通", "客户交流"],
            "项目管理": ["项目规划", "进度控制", "风险管理"]
        }
    }
```

3.2 项目经历质量评估

这块主要从项目描述完整性、贡献明显性和技术深度分析

代码块

```
project_quality_metrics = {
    "描述完整性": {
        "检查项": ["背景", "目标", "行动", "结果"],
        "STAR原则匹配度": "0-1评分"
    },
    "贡献明确性": {
        "指标": ["第一人称使用率", "具体动作动词", "量化描述"],
        "个人贡献指数": "基于模糊词检测"
    },
    "技术深度": {
        "层次": ["应用层", "原理层", "优化层", "架构层"],
        "深度评分": "基于技术讨论的复杂性"
    }
}
```

四、岗位匹配度

4.1 JD关键词匹配

这块主要提取JD中的关键词，接着与用户简历做对比

代码块

```
jd_matching_algorithm = {
    "硬技能匹配": {
        "提取方法": "从JD提取技术要求",
        "匹配算法": "余弦相似度 + 词向量",
        "权重分配": "核心技能×3，相关技能×1"
    },
    "经验匹配": {
        "维度": ["行业经验", "项目规模", "团队规模"],
        "匹配度计算": "基于年限和复杂度"
    },
    "软技能匹配": {
        "提取关键词": ["沟通能力", "团队合作", "解决问题"],
        "上下文分析": "从项目描述中推断"
    }
}
```

4.2 对比分析

对比完之后还要给出建议

代码块

```
gap_analysis = {
    "技能对比": {
        "识别方法": "JD要求 - 简历现有技能",
        "优先级排序": ["必须项", "加分项", "可选"],
        "学习建议": "推荐学习路径"
    },
    "经验对比": {
        "对比维度": ["项目复杂度", "管理经验", "业务规模"],
        "量化差距": "基于行业标准"
    }
}
```

五、问题生成

5.1 基于简历内容的问题生成

这一块就是四个梯级问题的生成，根据不同的关键词生出问题

代码块

```
question_generation_rules = {
    "L1基础破冰问题": {
        "生成逻辑": "针对简历列出的核心技术",
        "问题模板": "请解释{技术}的{原理/应用场景}",
        "生成数量": "每个核心技术1-2题"
    },
    "L2能力验证问题": {
        "选择策略": "选择最相关或描述最详细的1-2个项目",
        "问题维度": ["技术实现", "个人贡献", "成果验证"],
        "追问准备": "为每个问题准备2-3层追问"
    },
    "L3深度穿透问题": {
        "触发条件": "检测到夸大词或模糊表述",
        "问题类型": ["极端场景", "异常处理", "失败经历"],
        "设计原则": "逐渐增加压力，考察极限"
    },
    "L4综合潜力问题": {
        "生成依据": "基于岗位要求和行业趋势",
        "问题领域": ["技术趋势", "架构思考", "方法论"],
        "评估重点": "思维结构化和创新性"
    }
}
```

5.2 基于追问逻辑的问题生成

这一块看用户的回答是否模糊、是否不自信或太自信、是否太笼统了，进行追问

代码块

```
follow_up_strategies = {
    "模糊回答追问": {
        "触发词": ["大概", "通常", "一般", "可能"],
        "追问模板": "请具体说明{具体方面}，包括{细节要求}",
        "追问层次": ["具体细节", "数据支撑", "原理解释"]
    },
    "矛盾点追问": {
        "检测方法": "简历与回答不一致",
        "追问策略": "直接指出矛盾，要求解释",
        "风险评估": "根据解释合理性调整风险分数"
    },
    "技术深度追问": {
        "触发条件": "回答停留在表面",
        "追问方向": ["实现原理", "性能考量", "备选方案"],
        "深度控制": "根据候选人水平动态调整"
    }
}
```