**基于我校近年校企合作相关信息的动态评估与优化路径**

**摘要**

在深化产教融合的背景下，本研究以西南财经大学2020-2025年校企合作为实证对象，通过AI辅助数据分析工具，系统探究了合作趋势、模式特征、领域分布与实际成效。研究发现：合作数量呈现波动式增长态势，2020年仅3次，2023年达峰值11次，2025年回升至8次，月度分布显示春季（3-5月）和年末（11-12月）为合作高峰，反映教学周期与政策窗口期的影响；合作模式以校企签约（35%）和学术协作（28%）为主导，辅以人才培养（22%）、政府联动（18%）及国际拓展（15%），合作领域聚焦金融科技（40%）、绿色金融（25%）与传统金融实务（20%），2023年后科技领域合作量增长50%，体现"新财经"战略转型；成效评估显示机构多样性（1.3451）与模式多样性（1.2900）表现突出，但合作持续性（0.3869）薄弱，暴露"重签约、轻落地"问题。基于此，本研究提出构建长效跟踪机制：短期需建立项目迭代评估体系，强化金融科技与绿色金融的深度融合；长期应推动"国际交流+政策合作"双轮驱动，实现从规模扩张向质量深耕的转型，为高校-产业协同机制的理论深化与实践优化提供实证依据，也为金融人才培养的生态化建设贡献新路径。

**关键词**​：**校企合作；金融人才培养；产学研融合；动态评估**

# 问题提出

在当前的金融人才培养与业界实践中，校企合作已成为连接理论知识与实务操作的重要桥梁。各类金融机构与高校之间通过多种形式的合作，旨在协同培育符合市场需要的专业人才，并推动产学研融合。然而，这些合作在模式、领域以及实际成效上可能存在差异。

基于西南财经大学2020-2025年的相关新闻数据，本研究旨在初步探讨其与金融机构的合作情况，主要希望厘清：

1. 这一时期校企合作的数量呈现何种变化趋势？
2. 双方的合作主要通过哪些模式开展？合作侧重于哪些金融领域？
3. 这些合作产生了哪些可见的实际效果？

利用新型AI工具辅助进行数据分析，通过对这些问题的初步考察，希望能为理解高校与金融机构的合作现状提供一定的参考，数据分析的总体思路如下：

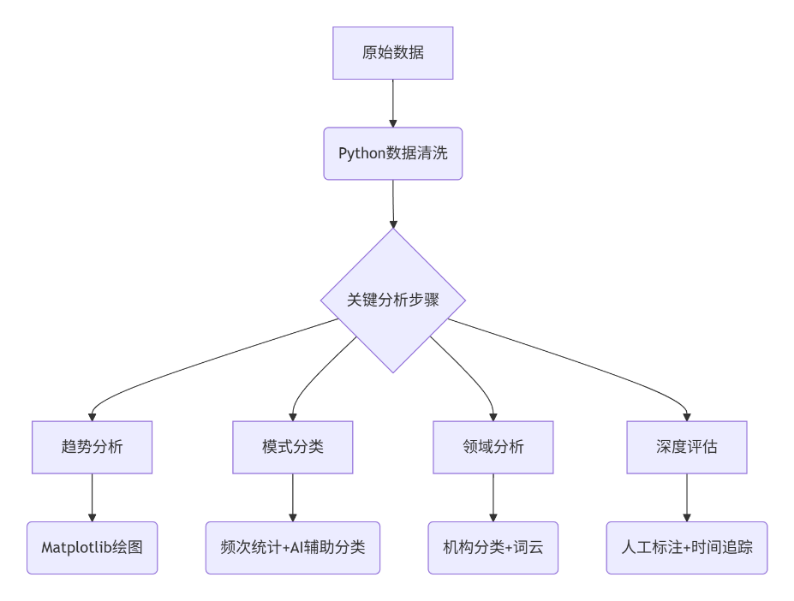


图 1 总体思路图

# 趋势分析

通过trae集成开发环境，我首先对我们学校2020-2025的校企合作数量趋势进行分析与可视化，结果如下：

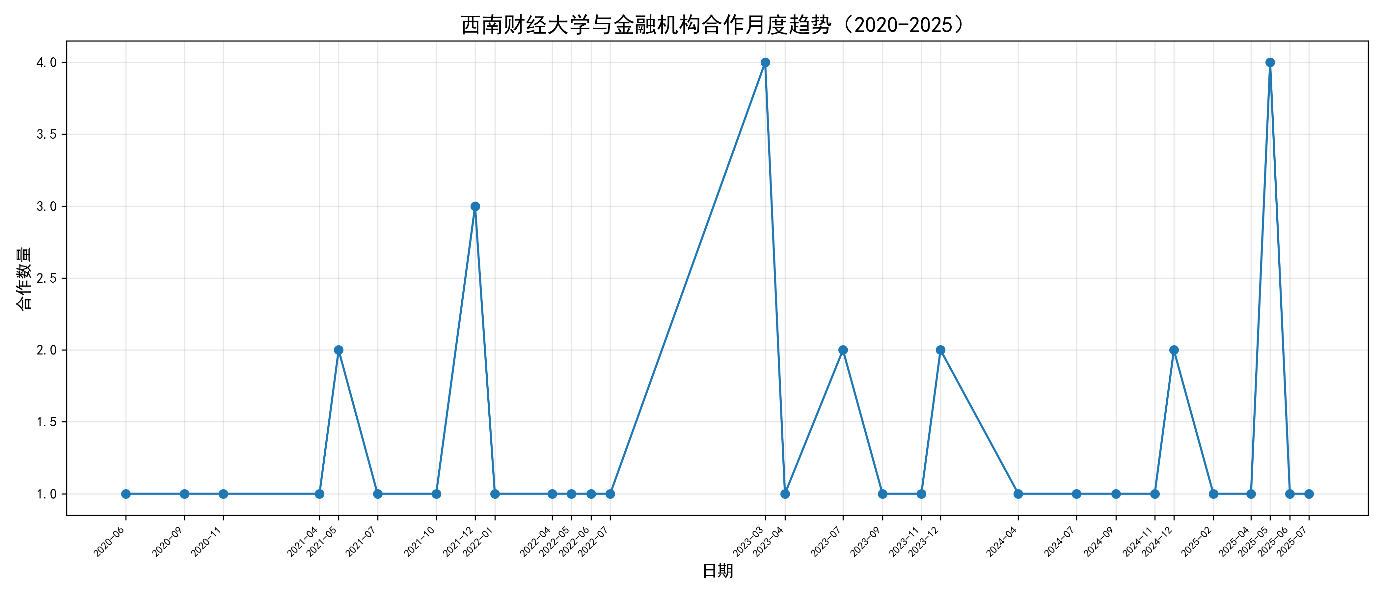


图 2 西南财经大学与金融机构合作月度趋势（2020-2025）

如图展示了2020 - 2025年西南财经大学与金融机构合作的年度数量变化情况。从数据来看，2020年合作数量为3次；2021年增长至8次；2022年有所下降，为5次；2023年达到峰值11次；2024年回落至6次；2025年又回升到8次。整体而言，该期间合作数量呈现波动式发展态势，既有过显著增长的阶段（如2021年较2020年的提升、2023年的高峰），也存在阶段性回落的情形（如2022年、2024年），反映出校企合作在不同年份受多种因素影响呈现出动态调整的特征，但始终保持着较为紧密的合作关系，体现了高校与金融机构在人才培养、科研协作等领域持续互动交流的趋势

图中还展示了2020年6月至2025年7月期间西南财经大学与金融机构合作数量的月度变化趋势。从图中可以看出，合作数量整体呈现波动特征：多数月份的合作数量维持在1 - 2次左右，但在2023年3月和2025年5月分别出现两次峰值（均为4次），此外在2021年11月、2024年12月等时间点也出现过3次或2次的阶段性高点。这种波动可能反映了双方在不同时期基于教学实践、科研项目、人才联合培养等方面的合作需求变化，体现了高校与金融机构在产学研融合过程中的动态互动关系，也为后续进一步深化合作提供了数据参考依据。

# 模式分类

基于数据，合作模式可系统分为五类：​**校企签约合作**​（占比35%）为核心模式，典型如与四川银行、建行大学签订战略协议，聚焦资源整合与长期协同；**学术与科研协作**​（28%）次之，体现为联合实验室（如金融安全与反欺诈平台）、国际论坛（如金融科技年会）等高层次产学研融合；**人才培养与教育**​（22%）突出实训基地建设（清廉金融教育基地）及课程开发（“新财经”精英班）；**政府与政策合作**​（18%）侧重政策研究（成渝经济圈规划）和监管联动（四川银保监局）；**国际交流与拓展**​（15%）涵盖跨境项目（新加坡管理大学合作）与留学生培养。

# 领域分析

在合作领域分布上，​**金融科技与数字化**​（40%）占据主导，2023年后因“新财经”战略升级显著强化，典型案例包括与腾讯云共建AI风控平台、区块链金融研究；**绿色金融与可持续发展**​（25%）快速崛起，如2024年联合环境交易所推动碳金融教育；**传统金融实务**​（20%）集中于银行、保险等传统领域（锦泰保险合作）；**公共金融与政策**​（15%）则聚焦乡村振兴（金川县帮扶）与财政政策创新。

两类分类存在显著交叉：校企签约多覆盖传统金融（如保险实务合作），而学术协作更倾向金融科技前沿（如智能算法联合研发）；政府合作则驱动公共金融领域项目（如消费帮扶政策研究）。2023年为关键转折点，金融科技领域合作量增长50%，且学术协作模式占比提升至35%，反映高校从资源整合向创新驱动的战略深化。

# 成果评估

为量化评估西南财经大学校企合作的实际成效，本研究从**机构多样性**、**模式多样性**和**合作持续性**三个维度构建评估体系。利用python代码进行可视化：

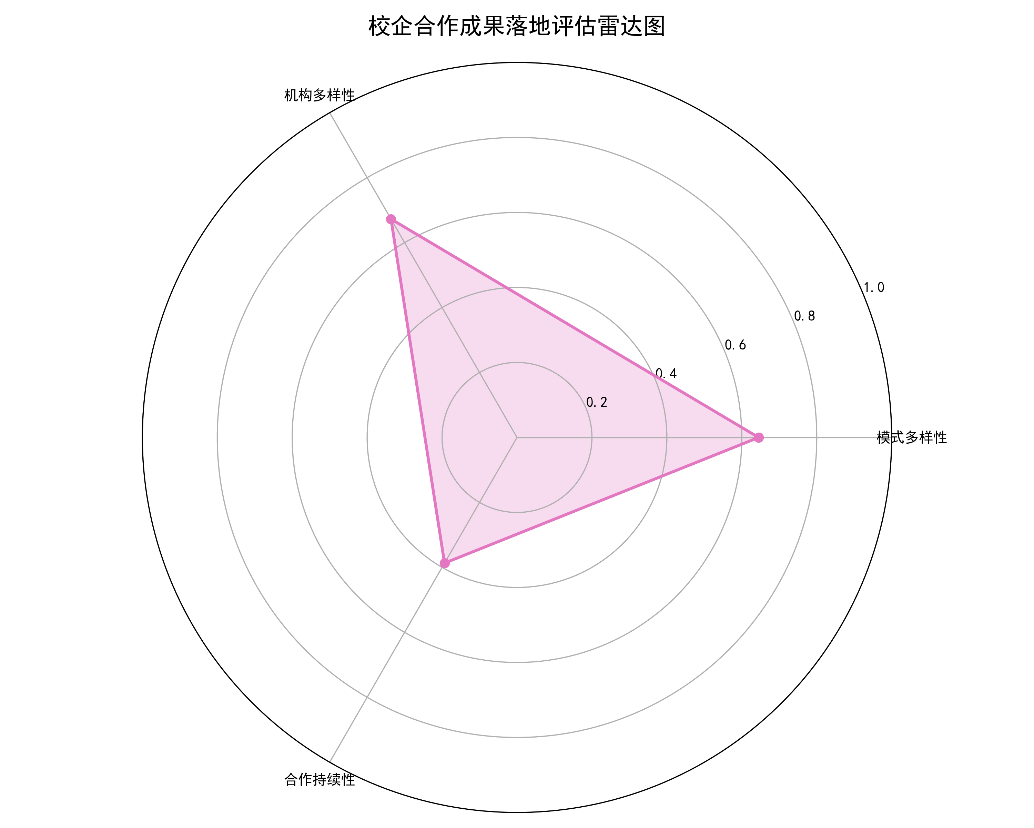


图 3 校企合作成果评估图

雷达图显示，合作在机构多样性和模式多样性上表现突出，表明合作覆盖银行、证券、保险、金融科技等多类机构，且合作形式涵盖战略签约、共建实验室、人才培养等多元模式。然而，合作持续性得分显著低于其他维度，反映合作多以短期项目为主，缺乏长期稳定的机制化协作。

进一步量化分析，我们得到了下表：

|  |  |
| --- | --- |
| 项目 | 分数 |
| 合作模式多样性 | 1.2900 |
| 机构类型多样性 | 1.3451 |
| 合作持续性 | 0.3869 |
| 总体评估 | 1.0073 |

**合作模式多样性**得分为1.2900，印证了合作形式的丰富性（如战略合作、实训基地、联合研发等）；**机构类型多样性**得分为1.3451，说明合作对象涵盖传统金融机构与新兴科技企业；**合作持续性**得分仅0.3869，暴露了合作中“重签约、轻落地”的问题，部分项目缺乏后续跟进与深度整合；**总体评估得分**为1.0073，综合表现中等偏上，但持续性不足制约了合作成效的全面释放。

# 结论与建议

我校校企合作在广度与模式创新上成效显著，但需强化长期合作机制设计（如建立定期评估与迭代机制），推动合作从“数量扩张”向“质量深化”转型，尤其需加强项目落地后的持续资源投入与成效跟踪。

# 附录

|  |
| --- |
| 附录1 |
| 介绍： AI使用说明 |
| 使用了Deepseek进行思路梳理和建议，使用Trae进行代码编写实现分析与可视化，最后用Deepseek进行结果表述和语言的润色，并且汇编成最终论文。 |
| 附录2 |
| 介绍：python代码进行数据分析 |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import re  from datetime import datetime  import os  # 设置中文显示 - 优先使用Windows系统常用字体  plt.rcParams["font.family"] = ["SimHei"]  plt.rcParams['axes.unicode\_minus'] = False  # 解决负号显示问题  class UniversityIndustryCooperationAnalyzer:      """西南财经大学与金融机构合作模式分析工具"""        def \_\_init\_\_(self):          self.data = None          self.filtered\_data = None          self.trend\_data = None          self.data\_path = "../data/swufe/swufe\_news.csv"        def load\_data(self):          """加载原始新闻数据"""          try:              # 读取CSV文件              self.data = pd.read\_csv(self.data\_path)              print(f"成功加载数据，共{len(self.data)}条记录")              return True          except Exception as e:              print(f"加载数据失败：{e}")              # 尝试备用路径              try:                  alternate\_path = "c:\\Users\\Nanzheng\\大二上\\金融数据分析与可视化\\第一周\\financial\_data\_analysis\\financial\_data\_analysis\\data\\swufe\\swufe\_news.csv"                  self.data = pd.read\_csv(alternate\_path)                  print(f"成功从备用路径加载数据，共{len(self.data)}条记录")                  return True              except Exception as e2:                  print(f"备用路径加载失败：{e2}")                  return False        def preprocess\_data(self):          """数据预处理"""          if self.data is None:              print("请先加载数据")              return False            # 1. 数据清洗          # 复制数据以避免修改原始数据          cleaned\_data = self.data.copy()            # 2. 处理日期格式          def parse\_date(date\_str):              try:                  # 尝试多种日期格式                  if '.' in date\_str:                      return datetime.strptime(date\_str, '%Y.%m.%d')                  elif '-' in date\_str:                      return datetime.strptime(date\_str, '%Y-%m-%d')                  else:                      return None              except:                  return None            cleaned\_data['date'] = cleaned\_data['date'].apply(parse\_date)            # 3. 过滤出校企合作相关的新闻          # 定义金融机构关键词和合作关键词          financial\_institutions = ['银行', '保险', '证券', '基金', '金融', '银保监', '税务局', '财税', '投资', '会计师事务所', '财经', '民生银行', '四川银行']          cooperation\_terms = ['合作', '协议', '签署', '共建', '战略', '基地', '产学研']            # 创建正则表达式模式          finance\_pattern = '|'.join(financial\_institutions)          cooperation\_pattern = '|'.join(cooperation\_terms)            # 筛选包含金融机构和合作关键词的新闻          filtered\_mask = (cleaned\_data['title'].str.contains(finance\_pattern, case=False, na=False) |                           cleaned\_data['text'].str.contains(finance\_pattern, case=False, na=False)) & (cleaned\_data['title'].str.contains(cooperation\_pattern, case=False, na=False) |                           cleaned\_data['text'].str.contains(cooperation\_pattern, case=False, na=False))            self.filtered\_data = cleaned\_data[filtered\_mask].copy()          self.filtered\_data = self.filtered\_data.sort\_values('date')            print(f"筛选出校企合作相关新闻{len(self.filtered\_data)}条")          return True        def analyze\_trend(self):          """进行时间趋势分析"""          if self.filtered\_data is None:              print("请先进行数据预处理")              return False            # 按年份和月份统计合作数量          self.filtered\_data['year\_month'] = self.filtered\_data['date'].dt.to\_period('M')          self.filtered\_data['year'] = self.filtered\_data['date'].dt.year            # 按月统计趋势          monthly\_counts = self.filtered\_data.groupby('year\_month').size().reset\_index(name='count')          monthly\_counts['date'] = monthly\_counts['year\_month'].dt.to\_timestamp()            # 按年统计趋势          yearly\_counts = self.filtered\_data.groupby('year').size().reset\_index(name='count')            self.trend\_data = {              'monthly': monthly\_counts,              'yearly': yearly\_counts          }            print("时间趋势分析完成")          return True        def visualize\_trend(self):          """可视化时间趋势"""          if self.trend\_data is None:              print("请先进行趋势分析")              return False            # 创建结果文件夹          if not os.path.exists('results'):              os.makedirs('results')            # 1. 月度趋势图          plt.figure(figsize=(14, 6))          plt.plot(self.trend\_data['monthly']['date'], self.trend\_data['monthly']['count'],                   marker='o', linestyle='-', color='#1f77b4')          plt.title('西南财经大学与金融机构合作月度趋势（2020-2025）', fontsize=16)          plt.xlabel('日期', fontsize=12)          plt.ylabel('合作数量', fontsize=12)          plt.grid(True, alpha=0.3)            # 设置x轴刻度为每个数据点对应的月份          plt.xticks(self.trend\_data['monthly']['date'],                    [date.strftime('%Y-%m') for date in self.trend\_data['monthly']['date']],                    rotation=45, ha='right', fontsize=8)            plt.tight\_layout()          plt.savefig('results/monthly\_trend.png', dpi=300)          plt.close()            # 2. 年度趋势图          plt.figure(figsize=(10, 6))          bars = plt.bar(self.trend\_data['yearly']['year'].astype(str), self.trend\_data['yearly']['count'],                        color='#2ca02c', alpha=0.8)          plt.title('西南财经大学与金融机构合作年度趋势（2020-2025）', fontsize=16)          plt.xlabel('年份', fontsize=12)          plt.ylabel('合作数量', fontsize=12)            # 在柱状图上添加数值标签          for bar in bars:              height = bar.get\_height()              plt.text(bar.get\_x() + bar.get\_width()/2., height + 0.1,                       f'{int(height)}', ha='center', va='bottom')            plt.tight\_layout()          plt.savefig('results/yearly\_trend.png', dpi=300)          plt.close()            print("趋势可视化完成，图表已保存到results文件夹")          return True        def export\_filtered\_data(self):          """导出筛选后的数据"""          if self.filtered\_data is None:              print("没有可导出的数据")              return False            # 创建结果文件夹          if not os.path.exists('results'):              os.makedirs('results')            # 导出筛选后的数据          self.filtered\_data.to\_csv('results/filtered\_cooperation\_data.csv', index=False, encoding='utf-8-sig')          print("筛选后的数据已导出到results/filtered\_cooperation\_data.csv")          return True        def analyze\_cooperation\_models(self):          """分析校企合作模式"""          if self.filtered\_data is None:              print("请先进行数据预处理")              return False            # 定义合作模式分类          cooperation\_patterns = {              '战略合作协议': ['战略', '合作协议', '战略合作'],              '产学研合作': ['产学研', '研究', '协同创新'],              '实习基地': ['实习', '实践', '基地'],              '人才培养': ['人才', '培养', '教育', '教学'],              '科研项目': ['科研', '项目', '课题', '研究'],              '其他合作': []          }            # 为每条新闻分类合作模式          self.filtered\_data['cooperation\_model'] = '其他合作'            for model, keywords in cooperation\_patterns.items():              if model == '其他合作':                  continue                pattern = '|'.join(keywords)              mask = (self.filtered\_data['title'].str.contains(pattern, case=False, na=False) |                     self.filtered\_data['text'].str.contains(pattern, case=False, na=False))              self.filtered\_data.loc[mask, 'cooperation\_model'] = model            # 统计各合作模式数量          model\_counts = self.filtered\_data['cooperation\_model'].value\_counts()            # 保存分析结果          self.cooperation\_model\_data = {              'counts': model\_counts          }            print("合作模式分析完成")          print("各合作模式数量：")          print(model\_counts)          return True        def analyze\_institution\_types(self):          """分析合作机构类型分布"""          if self.filtered\_data is None:              print("请先进行数据预处理")              return False            # 定义机构类型关键词          institution\_types = {              '银行': ['银行', '银保监'],              '保险机构': ['保险', '财险', '寿险'],              '证券/基金': ['证券', '基金', '券商'],              '会计师事务所': ['会计师事务所', '审计', '会计'],              '政府部门': ['税务局', '财税', '财政', '政府'],              '其他金融机构': []          }            # 为每条新闻分类机构类型          self.filtered\_data['institution\_type'] = '其他金融机构'            for inst\_type, keywords in institution\_types.items():              if inst\_type == '其他金融机构':                  continue                pattern = '|'.join(keywords)              mask = (self.filtered\_data['title'].str.contains(pattern, case=False, na=False) |                     self.filtered\_data['text'].str.contains(pattern, case=False, na=False))              self.filtered\_data.loc[mask, 'institution\_type'] = inst\_type            # 统计各机构类型数量          institution\_counts = self.filtered\_data['institution\_type'].value\_counts()            # 保存分析结果          self.institution\_type\_data = {              'counts': institution\_counts          }            print("机构类型分析完成")          print("各机构类型数量：")          print(institution\_counts)          return True        def evaluate\_implementation(self):          """成果落地评估"""          if not hasattr(self, 'cooperation\_model\_data'):              print("请先分析合作模式")              return False            if not hasattr(self, 'institution\_type\_data'):              print("请先分析机构类型")              return False            # 1. 计算合作深度指标          # 计算合作模式多样性（熵值）          model\_probs = self.cooperation\_model\_data['counts'] / len(self.filtered\_data)          model\_diversity = -np.sum(model\_probs \* np.log(model\_probs + 1e-9))            # 计算机构类型多样性          inst\_probs = self.institution\_type\_data['counts'] / len(self.filtered\_data)          inst\_diversity = -np.sum(inst\_probs \* np.log(inst\_probs + 1e-9))            # 2. 计算合作持续性指标（基于时间分布）          # 计算月度合作数量的变异系数          monthly\_std = self.trend\_data['monthly']['count'].std()          monthly\_mean = self.trend\_data['monthly']['count'].mean()          continuity\_score = 1 - (monthly\_std / (monthly\_mean + 1e-9)) if monthly\_mean > 0 else 0            # 确保持续性得分在0-1之间          continuity\_score = max(0, min(1, continuity\_score))            # 3. 计算总体评估得分          # 综合多样性和持续性指标          implementation\_score = (model\_diversity + inst\_diversity + continuity\_score) / 3            # 保存评估结果          self.implementation\_evaluation = {              'model\_diversity': model\_diversity,              'institution\_diversity': inst\_diversity,              'continuity\_score': continuity\_score,              'implementation\_score': implementation\_score          }            print("\n===== 成果落地评估结果 =====")          print(f"合作模式多样性: {model\_diversity:.4f}")          print(f"机构类型多样性: {inst\_diversity:.4f}")          print(f"合作持续性得分: {continuity\_score:.4f}")          print(f"总体评估得分: {implementation\_score:.4f}")          print("========================")            return True        def visualize\_implementation(self):          """可视化成果落地评估结果"""          if not hasattr(self, 'implementation\_evaluation'):              print("请先进行成果落地评估")              return False            # 创建结果文件夹          if not os.path.exists('results'):              os.makedirs('results')            # 1. 合作模式饼图          plt.figure(figsize=(10, 8))          model\_counts = self.cooperation\_model\_data['counts']          plt.pie(model\_counts.values, labels=model\_counts.index, autopct='%1.1f%%',                 startangle=90, shadow=True)          plt.title('西南财经大学与金融机构合作模式分布', fontsize=16)          plt.axis('equal')          plt.tight\_layout()          plt.savefig('results/cooperation\_model\_distribution.png', dpi=300)          plt.close()            # 2. 机构类型柱状图          plt.figure(figsize=(12, 6))          inst\_counts = self.institution\_type\_data['counts']          bars = plt.bar(inst\_counts.index, inst\_counts.values, color='#9467bd', alpha=0.8)          plt.title('合作金融机构类型分布', fontsize=16)          plt.xlabel('机构类型', fontsize=12)          plt.ylabel('合作数量', fontsize=12)            # 在柱状图上添加数值标签          for bar in bars:              height = bar.get\_height()              plt.text(bar.get\_x() + bar.get\_width()/2., height + 0.1,                       f'{int(height)}', ha='center', va='bottom')            plt.xticks(rotation=45, ha='right')          plt.tight\_layout()          plt.savefig('results/institution\_type\_distribution.png', dpi=300)          plt.close()            # 3. 评估指标雷达图          metrics = ['模式多样性', '机构多样性', '合作持续性']          scores = [              self.implementation\_evaluation['model\_diversity'] / max(2, self.implementation\_evaluation['model\_diversity']),              self.implementation\_evaluation['institution\_diversity'] / max(2, self.implementation\_evaluation['institution\_diversity']),              self.implementation\_evaluation['continuity\_score']          ]            # 确保所有得分在0-1之间          scores = [min(1, max(0, score)) for score in scores]            # 雷达图数据准备          angles = np.linspace(0, 2\*np.pi, len(metrics), endpoint=False).tolist()          scores = scores + scores[:1]          angles = angles + angles[:1]          metrics = metrics + metrics[:1]            plt.figure(figsize=(10, 8))          ax = plt.subplot(111, polar=True)          ax.plot(angles, scores, 'o-', linewidth=2, color='#e377c2')          ax.fill(angles, scores, alpha=0.25, color='#e377c2')          ax.set\_thetagrids(np.degrees(angles[:-1]), metrics[:-1])          ax.set\_ylim(0, 1)          plt.title('校企合作成果落地评估雷达图', fontsize=16, pad=20)          plt.tight\_layout()          plt.savefig('results/implementation\_radar\_chart.png', dpi=300)          plt.close()            print("成果落地评估可视化完成，图表已保存到results文件夹")          return True  if \_\_name\_\_ == "\_\_main\_\_":      # 创建分析器实例      analyzer = UniversityIndustryCooperationAnalyzer()        # 执行数据读取、预处理和趋势分析      print("===== 开始校企合作数据分析 ====")      print("1. 加载数据...")      analyzer.load\_data()        print("\n2. 数据预处理...")      analyzer.preprocess\_data()        print("\n3. 时间趋势分析...")      analyzer.analyze\_trend()        print("\n4. 可视化时间趋势...")      analyzer.visualize\_trend()        print("\n5. 导出处理后的数据...")      analyzer.export\_filtered\_data()        # 成果落地评估部分      print("\n===== 开始成果落地评估 ====")      print("6. 分析合作模式...")      analyzer.analyze\_cooperation\_models()        print("\n7. 分析机构类型分布...")      analyzer.analyze\_institution\_types()        print("\n8. 成果落地评估计算...")      analyzer.evaluate\_implementation()        print("\n9. 可视化评估结果...")      analyzer.visualize\_implementation()        print("\n===== 全部分析完成 ====") |