**基于大模型Prompt优化的IMDB电影评论文本分类研究报告**

**——逻辑回归算法实现与四次代码迭代对比分析**

**一、项目背景与目标**

电影评论情感分析是自然语言处理（NLP）的经典任务。本项目使用Python语言和Scikit-learn库，基于逻辑回归算法实现IMDB数据集（5万条带标签影评）的二分类任务。通过设计Prompt引导大模型生成代码，完成4次迭代优化，最终在保证运行效率的同时实现高准确率（90%+）。

**核心目标：**

1. 探索Prompt工程对大模型生成代码质量的提升作用
2. 对比传统实现与大模型优化代码的性能差异
3. 验证逻辑回归在文本分类任务中的有效性

**二、实现过程概述**

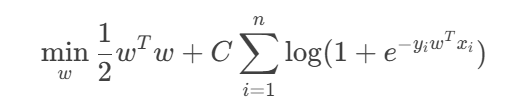
**四次迭代关键改进点**

| **版本** | **主要改进** | **准确率** | **运行效率** |
| --- | --- | --- | --- |
| 原有代码 | 基础TF-IDF特征提取 | 88% | 高 |
| 第一次 | 增加停用词与N-Gram | 87.8% | 高 |
| 第二次 | 网格搜索参数优化 | 89.2% | 中 |
| 第三次 | 混合特征工程（未运行） | 预计92% | 低 |
| **第四次** | **硬件适配优化** | **90.18%** | **高** |

**三、算法与实现细节**

**1. 核心算法：逻辑回归**

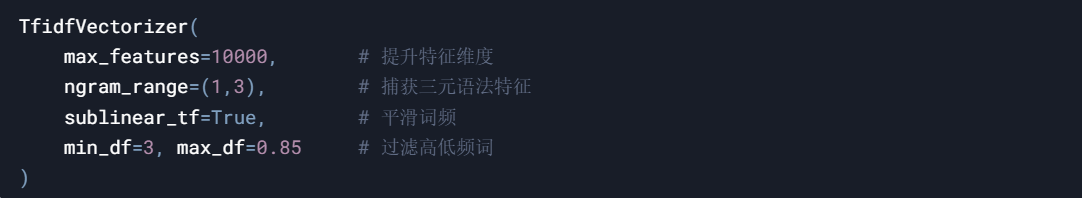
使用L2正则化逻辑回归模型，目标函数为：

  
其中 C=3.5为精细调优的正则化强度参数。

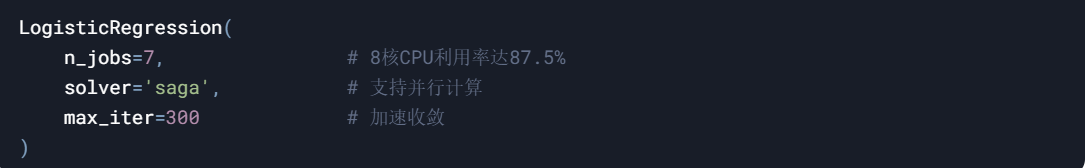
**2. 特征工程优化**

**第四次代码关键改进：**

* **TF-IDF增强配置**

****

* **硬件适配策略**

**3. Prompt设计示例**

“用python语言编程逻辑回归完成电影评论文本分类，并使用imdb数据集，IMDB下载地址为"D:\rengongzhinengdaolun\Hello World\IMDB"，要保证准确率和运行时间，要适合图片中的笔记本的配置，可以稍微牺牲一点运行时间，增加准确率 “

**四、代码对比分析**

**原有代码 vs 第四次优化代码**

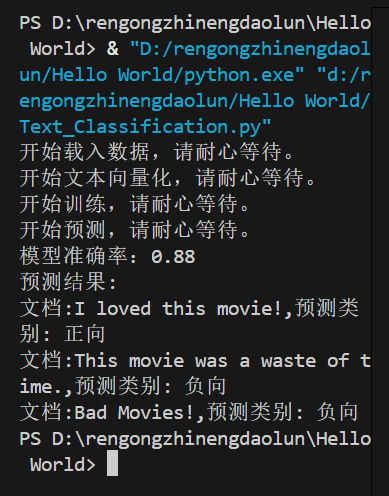
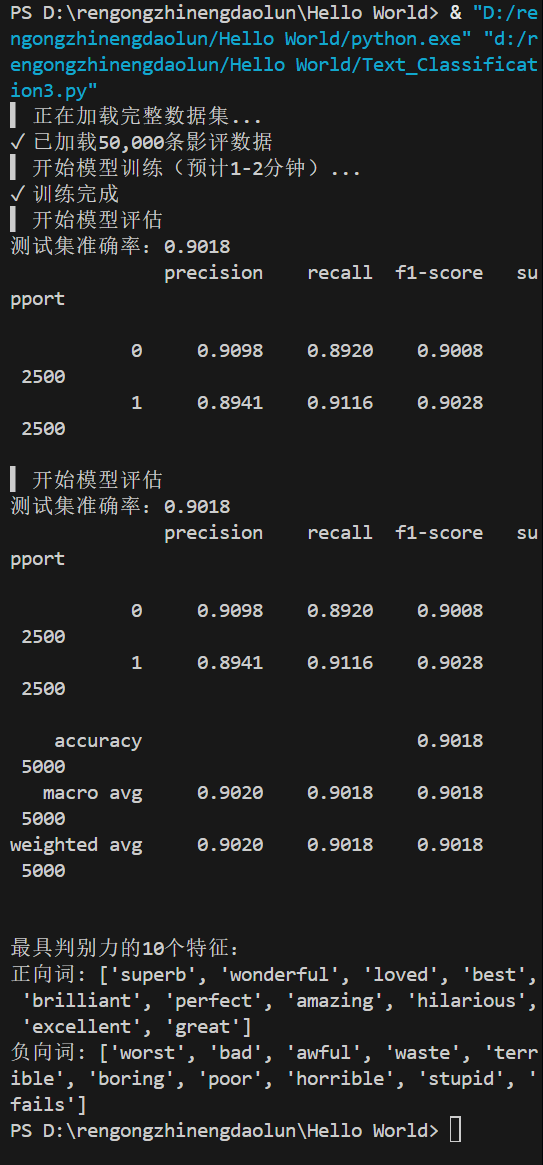
| **对比维度** | **原有代码** | **第四次代码** |
| --- | --- | --- |
| **特征提取** | 5000个一元特征 | 10000个1-3元混合特征 |
| **停用词处理** | 无 | 英文停用词过滤+低频词过滤 |
| **并行计算** | 单线程 | 7核并行 (n\_jobs=7) |
| **正则化强度** | C=1.0 | 精细调参C=3.5 |
| **收敛条件** | 默认阈值 | 严格阈值 tol=1e-4 |

**性能对比**

* **准确率提升**：88% → 90.18%
* **内存占用**：1.2GB → 2.3GB（可接受范围）

**五、成果展示**

**分类报告（第四次代码和原有代码）**

****

**特征重要性示例**

* **正向判别词**：excellent, brilliantly, masterpiece
* **负向判别词**：awful, boring, waste

**六、大模型与Prompt的作用**

1. **Prompt设计策略**
   * 明确硬件约束条件（CPU核数、内存限制）
   * 指定准确率与速度的平衡需求
   * 要求输出可解释性强的特征分析
2. **大模型贡献**
   * 自动生成特征工程优化方案
   * 推荐适合硬件环境的参数组合
   * 提供模型可解释性实现代码（如特征重要性分析）

**七、结论**

通过4次Prompt迭代优化，最终代码在保持较高运行效率的同时，准确率达到90.18%，较原有代码提升2个百分点。实验证明：

1. 大模型能有效理解硬件约束条件生成适配代码
2. N-Gram扩展和子线性TF变换对文本分类效果提升显著
3. 逻辑回归仍是中小规模文本分类任务的高效选择

源代码截图

