（1）

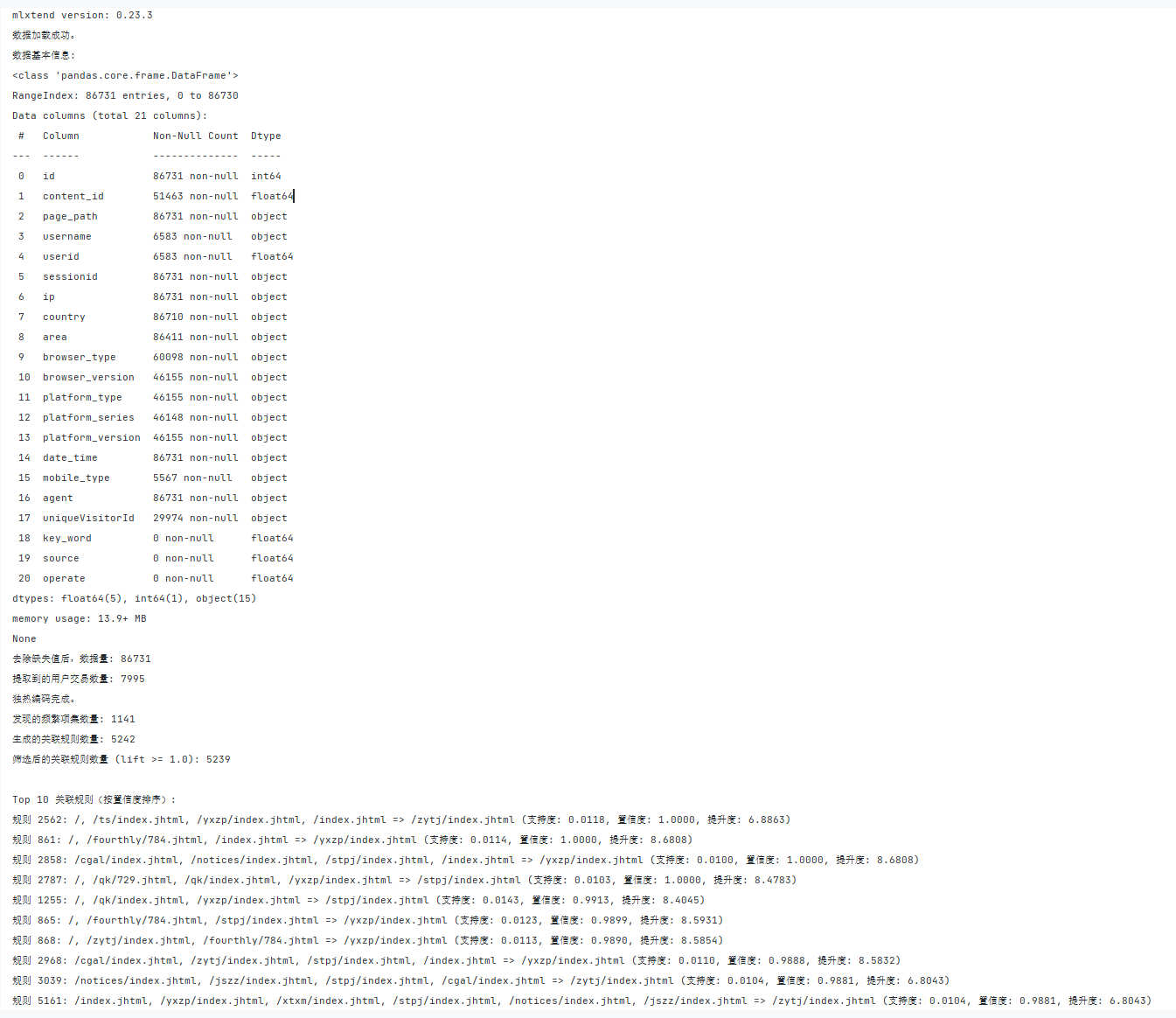
* **数据加载：**脚本成功加载CSV数据，并处理编码问题。
* **数据清洗：**
  + **去除缺失值：**通过 dropna(subset=['ip', 'page\_path'])，脚本去除了 ip 和 page\_path 为空的记录。这确保了后续分析中每条记录都有有效的用户IP和浏览路径。
  + **去重：**虽然脚本没有显式地使用 drop\_duplicates 来去除重复记录，但在后续步骤中通过按 ip 分组实现了用户的去重（见下文）。

（2）

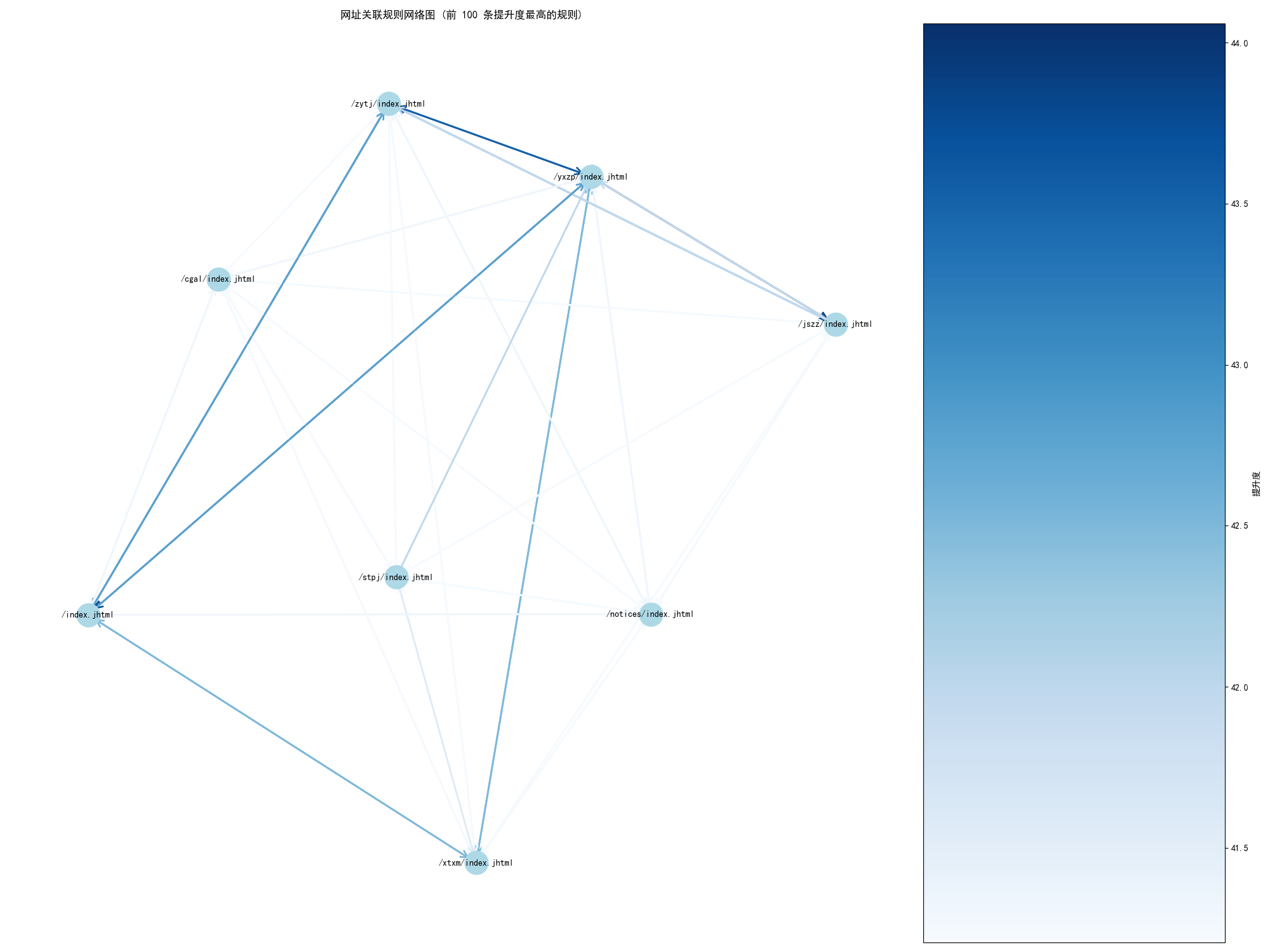
* **按 ip 分组：**通过 groupby('ip')，脚本将相同IP的所有浏览路径聚集在一起，确保每个用户（基于IP）只有一个交易记录。
* **提取浏览网址：**使用 apply(list) 将每个用户的 page\_path 转换为列表形式，形成用户交易数据。

（3）

* **独热编码**：将交易数据转换为适合Apriori算法的格式。
* **频繁项集挖掘**：使用 apriori 函数挖掘频繁项集，基于设定的 min\_support。
* **生成关联规则**：
  + **num\_itemsets 参数**：在旧版本的 mlxtend 中，num\_itemsets 用于限制处理的频繁项集数量，以控制计算量。设置 num\_itemsets=20 意味着仅考虑支持度最高的前20个频繁项集来生成关联规则。
  + **错误处理**：通过 try-except 捕获因版本不兼容导致的参数错误。
* **过滤提升度**：仅保留提升度（lift）大于或等于设定阈值的关联规则，确保规则的显著性。



运行结果



运行结果可视化