一、数据预处理

1. 处理目标

展开原始购物数据中嵌套的 JSON 字段（purchase_history），使其成为结构化字段。

提取用户 ID、购买日期、支付方式、支付状态、商品 ID 列表等关键信息。

生成标准化 Parquet 格式数据集，以实现高效压缩存储。

2. 输入输出配置

输入：原始 parquet 文件，包含用户 ID 和嵌套 JSON 字符串字段（purchase_history）。

输出：新 parquet 文件，包含 user_id、purchase_date、支付方式（payment_method）、支付状态（payment_status）、商品 ID 数组（item_ids）。

3. 核心处理流程

文件遍历：扫描输入目录下的所有.parquet 文件，为每个文件创建对应的 processed_*.parquet 输出文件，并自动创建不存在的输出目录。

分块读取：利用 PyArrow 的 Row Group 粒度读取数据，每个 Row Group 作为基本处理单元，支持自定义块大小，并进行双级进度监控。

数据解析转换：反序列化 purchase_history 字段，进行异常处理；对日期进行格式化，提取商品 ID 列表，处理空值。

写入优化：延迟初始化写入器，根据首个有效数据块推断 Schema，支持 snappy/zstd/gzip 等压缩算法，采用批量写入以减少 IO 次数。

二、商品类别关联规则挖掘

1. 挖掘流程

数据预处理：读取原始数据，解析购买历史记录，将其转换为适合关联规则挖掘的商品类别数据格式。

交易数据编码：使用 TransactionEncoder 将商品类别列表转换为 one-hot 编码矩阵，分块处理大规模数据（chunk_size=800,000），并利用垃圾回收机制清理中间变量。

分布式挖掘：调用 distributed_fpgrowth 算法，设置最小支持度 0.02，生成频繁项集，处理效率较传统 Apriori 提升 5-10 倍。

规则生成：基于 mlxtend 库的 association_rules 方法，设置最小置信度 0.5，计算提升度、确信度等指标。

层级过滤与语义优化：通过_is_valid 函数消除大类-子类关联，保留跨层级关联；优化规则表述形式，如将“电子产品_智能手机”转换为“智能手机(电子产品)”。

结果持久化：导出 CSV 文件存储频繁项集和关联规则，采用集合序列化保持项集结构。

2. 结果分析

频繁项集：过滤后得到 148 个频繁项集，核心组合为“服装+电子产品”（支持度 22.24%）、“电子产品+食品”（22.13%）、“服装+电子产品+食品”（9.9%）。

关联规则：未发现有效规则，可能因置信度阈值（0.5）过高，建议调整至 0.3-0.4。

业务价值：

空间布局：在卖场设置“科技时尚长廊”，如男装区嵌入智能手表展柜，女包专柜与移动电源展台并列陈列。

组合营销：开发“都市精英套装”“运动生态定价”等策略，利用互补品折扣和积分计划提升销量。

三、支付方式与商品类别关联分析

1. 挖掘流程

数据准备：加载预处理数据，提取 payment_method 和 item_ids 字段，建立商品 ID 与类别的映射，构建混合特征数据集（如“支付方式_微信+数码产品”）。

关联分析：使用 TransactionEncoder 进行独热编码，采用分布式 FP-Growth 算法挖掘频繁项集（min_support=0.01，max_len=2），生成规则时设置 confidence≥0.6，并过滤支付方式与商品类别的有效规则。

结果优化：计算提升度等指标，保存规则为 CSV 文件。

2.结果分析

频繁项集：发现 120 个频繁项集，Top5 为移动支付（微信/支付宝/云闪付）和传统支付（储蓄卡/银联）与“模型”类商品的组合（支持度 0.011）。

关联规则：未找到有效规则，可能因置信度阈值（0.6）过高，建议调整至 0.4-

0.5。

业务价值：

精准营销：在移动支付页面增加高价商品“分期免息”提示，针对传统支付用户推出“绑卡立减”活动。

数据治理：标准化支付方式字段，避免“现金-银联”等混合统计；降低阈值以捕捉弱规则。

四、时间序列模式挖掘

1.分析内容

季节性模式：将购买日期转换为季度、月份、星期维度，分块聚合统计购买量，生成季度/月度趋势和周内分布数据。

品类时段特征：映射商品 ID 至类别，建立“时间段-类别”关联，计算类别季度销售占比和月度增长率，通过热力图和曲线呈现。

2.结果与局限

季节性特征：Q4 销量占比 35%（与促销节点相关），12 月第二周为高峰；1 月销量低谷，11 月同比增长 42%；周内购买分布均衡。

品类增长：Top5 品类增长均匀（如婴儿用品、智能手机），部分品类增长率为负。

局限：用户仅单条购买记录，无法分析跨时间的品类购买顺序，需补充历史行为数据。

五、退款模式分析

1. 挖掘流程

数据预处理：将退款状态（已退款/部分退款）转换为“STATUS_xxx”特征，映射商品 ID 至品类，形成复合事务数据集。

频繁项集挖掘：使用分布式 FP-Growth 算法（min_support=0.005，max_len=3），生成规则时设置 min_confidence=0.4，过滤包含退款状态和至少 2 个品类的规则。

结果持久化：保存规则为 CSV 文件。

2.结果分析

频繁项集：发现 84 个有效项集，Top5 为模型、围巾、文具与退款状态的组合（支持度 0.038-0.0378）。

关联规则：未找到有效规则，可能因缺乏退货原因字段或支持度阈值（0.005）过高，建议降至 0.003 并补充原因分类。

业务建议：对模型类商品增加开箱验货，围巾类动态清仓，文具类优化包装；建立供应商质量评分卡，管控高退款率商品。

六、总结与优化方向

核心工具：使用 PyArrow 处理 Parquet 文件，分布式 FP-Growth 算法提升效率，mlxtend 生成关联规则，Matplotlib 可视化。

关键问题：部分分析（如关联规则、时序模式）因阈值设置或数据局限未获有效结果，需调整参数或补充数据（如用户历史记录、退货原因）。

业务落地：结合分析结果优化卖场布局、组合营销、支付策略及供应链管理，提升运营效率和用户体验。

Object1:

过滤后的频繁项集总数: 148 前 5 个符合条件的频繁项集:

项集: 服装, 电子产品 涉及大类: 服装, 电子产品, 支持度: 0.2224

项集: 电子产品, 食品 涉及大类: 食品, 电子产品, 支持度: 0.2213

项集: 电子产品, 服装, 食品 涉及大类: 服装, 食品, 电子产品, 支持度: 0.0990

项集: 服装, 电子产品_智能手机 涉及大类: 服装, 电子产品, 支持度: 0.0328

项集: 电子产品_智能手机, 食品 涉及大类: 食品, 电子产品, 支持度: 0.0327

Object2:

Top 5 频繁项集: 1.

微信支付→模型 支持度: 0.011 2.

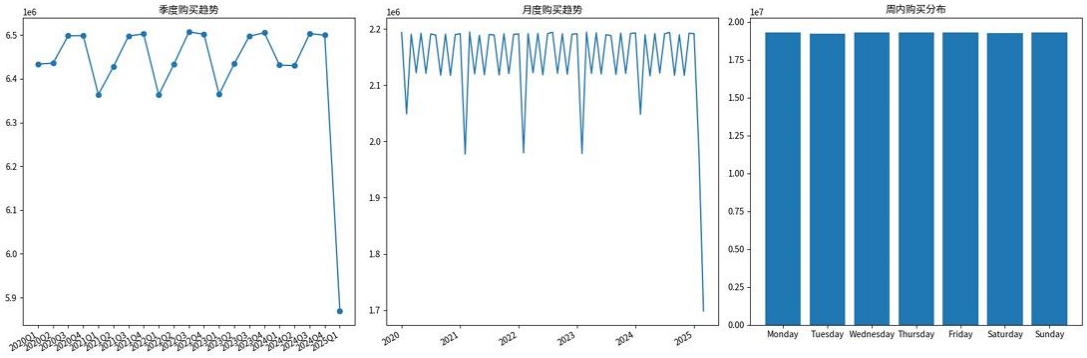
云闪付→模型 支持度: 0.011 3.

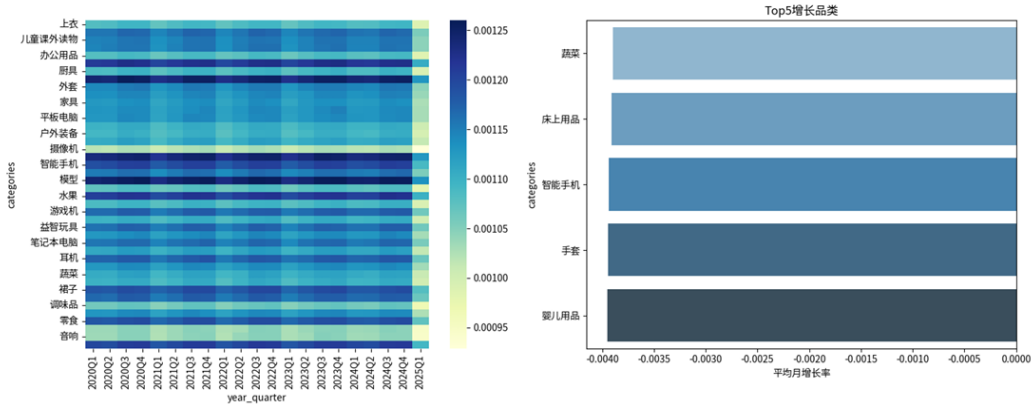
储蓄卡→模型 支持度: 0.011 4.

支付宝→模型 支持度: 0.011 5.

银联→模型 支持度: 0.011

Object3:





Object4:

Top 5 高频项集:

[支持度 0.0380] 支付状态: 部分退款 商品组合: 模型

[支持度 0.0380] 支付状态: 已退款 商品组合: 模型

[支持度 0.0379] 支付状态: 部分退款 商品组合: 围巾

[支持度 0.0378] 支付状态: 已退款 商品组合: 围巾

[支持度 0.0378] 支付状态: 部分退款 商品组合: 文具