

新方案相较于旧方案的优势在于以下几点：

1.

技术栈更新：新方案采用了更为流行和实用的技术栈，如Spark、PySpark、Numpy、Pandas、Python3、Vue、Bootstrap、ECharts等。这些技术在处理大数据、分析和可视化方面有更好的性能和易用性，同时也有更丰富的社区资源和支持。

2.

算法优化：新方案采用了基于Spark的并行化算法，如ALS（协同过滤算法）、KMeans、GBClassifier（Spark版本的XGBoost实现子集）。这些算法能够在大数据场景下高效地进行计算，提高整个系统的处理速度。

3.

数据处理与加载：新方案使用Spark读取CSV文件，可以从Hadoop或者文件系统中读取数据，提高了数据加载的灵活性。同时，新方案在数据处理和特征提取方面也进行了优化，使得数据分析更加精确和高效。

4.

可视化效果：新方案采用了Seaborn库和ECharts库进行数据可视化，提供了丰富的图表类型和美观的界面效果。前端采用Vue框架编写，与后端Flask进行交互，使得数据展示更加直观和动态。

5.

系统扩展性：新方案采用了前后端分离的架构，前端和后端可以独立进行开发和部署。同时，新方案的技术栈在未来仍具有较强的生命周期，有利于系统的长期维护和升级。

旧有方案存在的不足之处如下：

技术栈陈旧：旧方案采用了一些较为陈旧的技术，如Hive、HBase、ClickHouse和ElasticSearch。虽然这些技术在当时是适用的，但随着技术的发展，已经有了更为先进和易用的技术来替代它们。

算法效率较低：旧方案采用的FM、XGBoost、TF-IDF等推荐算法在处理大数据量的情况下可能效率较低，尤其在分布式计算环境中可能不够高效。

数据处理与加载不够灵活：旧方案在数据处理和加载方面相对较为繁琐，例如需要配置Hive仓库、HBase数据库等。这些操作可能会导致整个系统的复杂度上升，增加系统的维护成本。

可视化效果较差：旧方案在可视化方面可能存在局限性，只提供了折线图、表格等基本的展示形式，无法满足复杂和多样化的数据展示需求。

系统扩展性和可维护性较差：旧方案采用的技术栈和架构可能在扩展性和可维护性方面存在局限。随着业务的发展和技术的更新，旧方案可能难以适应新的需求，增加了系统的迭代和升级成本。

用户交互体验较差：旧方案在用户交互方面可能较为单一，没有提供丰富的API接口，使得用户与系统的交互较为不便。

总的说就是：

旧方案在技术选型、算法效率、数据处理与加载、可视化效果、系统扩展性和用户交互体验等方面存在不少不足之处。

这些问题可能会影响整个系统的性能、可用性和易用性，从而导致整个项目的竞争力下降。因此，采用新方案对于提高项目的整体质量具有重要意义。

新方案在技术选型、算法优化、数据处理与加载、可视化效果、系统扩展性和用户交互体验等方面相较于旧方案都有较大的优势。

这使得新方案在处理大数据推荐场景下更具竞争力，同时也有利于项目的长期维护和发展。