Melhorias na atualização de dados para evitar atrasos

Os atrasos na atualização de dados podem estar causando erros nos relatórios de previsão de fluxo de caixa. Para resolver este problema, seguem sugestões de melhorias:

1 – Automatização da coleta de dados

* Automatizar a integração entre os sistemas financeiros (ex.: ERP, APIs externas) e a base de dados usada pelo algoritmo.
* Utilizar ferramentas para agendar a coleta e sincronização de dados em tempo real ou em intervalos curtos (ex.: a cada hora).
* Garantir que qualquer transação registrada seja refletida na base de dados de previsão automaticamente.
* Reduz a dependência de processos manuais.
* Melhora a consistência e atualidade das informações.

2 – Sincronização em tempo real

* Implementar um mecanismo de sincronização contínua para captar transações no momento em que ocorrem.
* Configurar um sistema de eventos (ex.: Webhooks) que notifique o banco de dados quando uma nova transação for registrada no sistema financeiro.
* Integrar sistemas financeiros com tecnologia de mensageria (ex.: RabbitMQ, Kafka) para processar as atualizações em tempo real.
* Elimina o risco de atrasos na inclusão de transações.
* Permite maior precisão nos relatórios de previsão.

3 – Monitoramento de dados e alertas proativos

* Criar mecanismos para monitorar a sincronização de dados e alertar em caso de falhas.
* Implementar ferramentas de para observar os dados, como dashboards com métricas de sincronização e alertas automáticos em caso de falha.
* Configurar alertas via e-mail ou sistema de notificação (ex.: Slack, Teams) para informar atrasos ou erros na coleta de dados.
* Detecta e resolve problemas antes que afetem os relatórios.
* Garante maior confiabilidade no processo.

4 – Revisão de infraestrutura

* Avaliar e melhorar a infraestrutura de TI para lidar com o aumento no volume de transações e garantir desempenho.
* Identificar gargalos no sistema (ex.: consultas lentas no banco de dados).
* Migrar para serviços de nuvem ou banco de dados escaláveis (ex.: Oracle Cloud, AWS RDS).
* Implementar índices e particionamento no banco de dados para melhorar a performance.
* Aumenta a eficiência do sistema.
* Garante a escalabilidade em casos de aumento no volume de dados.