Equipe

Alex Assis - 10444718

João Pedro - 10444884

Kayque Mendes - 10444510

Laís Caniato - 10444191

Leonardo Toschi - 10444888

Definição de problema

Tendo em vista os prejuízos financeiros e danos de imagem decorrentes de atividades fraudulentas, a detecção de fraude se tornou de suma importância. A falta de medidas de proteção a fraude pode levar a experiências negativas dos clientes. Um cliente insatisfeito pode migrar para a concorrência e ainda fazer propaganda negativa, prejudicando a reputação da mesma. Estudos revelam que um cliente insatisfeito comunica essa insatisfação para 9 pessoas em média (Stichler e Schumacher, 2003)

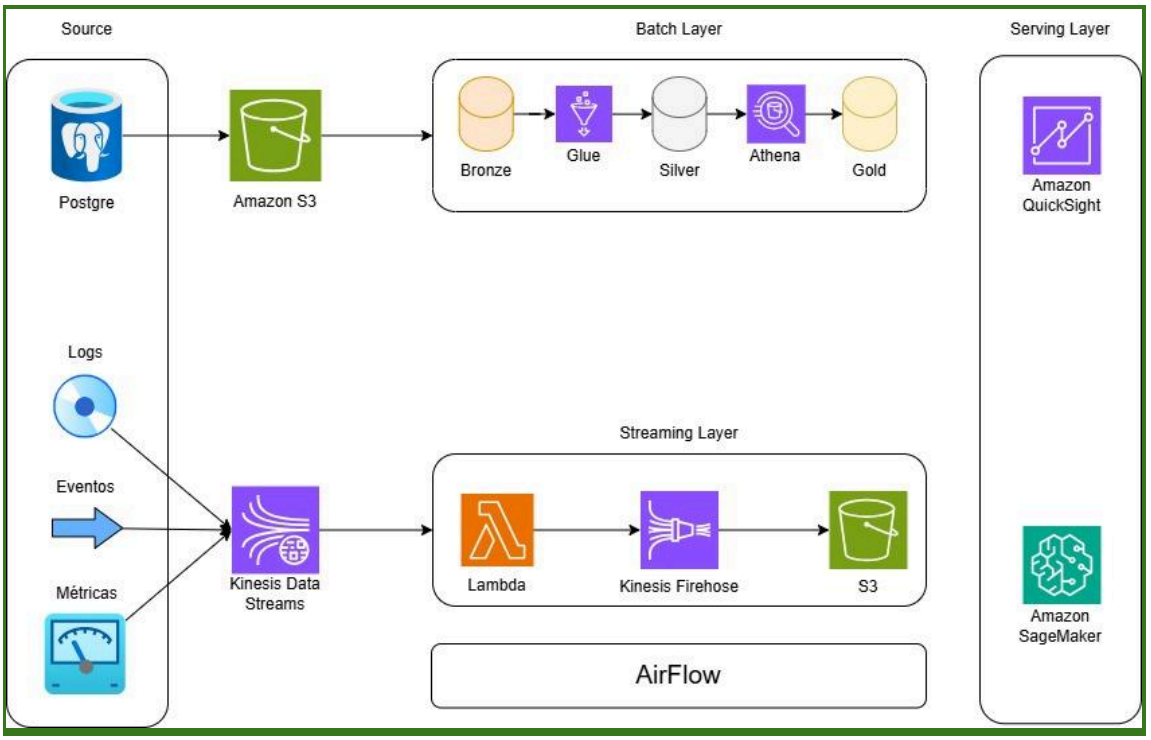
Além disso, segundo o STF, os bancos têm o dever de evitar fraudes, identificando e impedindo transações que não condizem com o perfil do cliente e caso não cumpram as medidas de prevenção, podem enfrentar ordens judiciais, que geralmente resultam em penalidades e multas. Um exemplo disso é o Bank of America que teve uma falha em um sistema de detecção de fraude e foi multado pelas agências reguladoras federais dos EUA em US$ 225 milhões.

E por isso, a empresa deseja detectar transações financeiras fraudulentas em tempo real, identificando e definindo comportamentos e padrões anômalos nas transações financeiras que indicam atividades fraudulentas (como roubo de identidade, transações não autorizadas, manipulação de dados, entre outros), visando garantir a segurança e satisfação de nossos clientes e evitando prejuízos financeiros.

Escolha da base

Para definir a nossa base de dados a ser utilizada para o projeto, utilizamos a ferramenta Kaggle, que fornece inúmeras bases de dados de maneira pública, que podem ser instaladas para manipulação como o usuário desejar.
Buscamos por uma base de dados relacionada a atividades fraudulentas em bancos (https://www.kaggle.com/code/miguelsevero/detec-o-de-transa-es-fraudulentas-ml/) e a utilizamos para elaboração da solução.

Desenho da arquitetura



Stack de Ferramentas

Para desenvolver a arquitetura na nuvem AWS para detecção de fraudes em uma instituição financeira, são necessárias ferramentas capazes de processar grande volume de dados, tanto históricos quanto em tempo real. Pensando nisso foram selecionadas as seguintes ferramentas:

Categoria	Serviços	Descrição
Ingestão	**Amazon Kinesis Data Streams**	Ingestão de dados em tempo real das transações bancárias, configurado com auto scaling para escalabilidade (speed layer)
	AWS Glue	Ingestão de dados históricos ou arquivos CSV (Batch Layer)
Processamento	**AWS Lambda**	Processamento de lógica customizada de detecção de fraudes em tempo real (Speed Layer), configurado com auto scaling para escalabilidade

	Amazon EMR	Processamento de dados com frameworks de big data (Batch Layer), como Apache Spark, Hive ou Presto, sem necessidade de gerenciar clusters manualmente
Armazenamento	**Amazon S3**	Armazenamento central para dados históricos
	Amazon Redshift	Armazenamento e análise de dados históricos para relatórios e aprendizado de máquina
	Amazon DynamoDB	Banco de dados NoSQL para armazenar eventos recentes e resultados de análises em tempo real, configurado com auto scaling para escalabilidade
Modelagem e Machine Learning	**Amazon SageMaker**	Treinamento e implementação de modelos de machine learning baseados em dados históricos armazenados na Batch Layer
	SageMaker Model Monitor	Monitoramento de performance dos modelos em tempo real
Detecção e Alertas	**Amazon EventBridge**	Acionamento de eventos baseados em regras (por exemplo, transações suspeitas)
	Amazon SNS (Simple Notification Service)	Envio de notificações de alertas (e-mails, mensagens ou integração com sistemas externos)
	Amazon CloudWatch	Monitoramento, logging de métricas e alertas
Governança	**AWS Step Functions**	Orquestração de workflows entre a Batch Layer e a Speed Layer

	AWS Lake Formation	Gerenciamento e proteção de acesso aos dados
	AWS IAM	Controle granular de acesso aos serviços e dados
Custo	**AWS Cost Explorer**	Monitoramento de gastos e otimização de recursos
Segurança	**AWS Key Management Service (KMS)**	Criptografia de dados
	AWS Shield	Proteção contra ataques DDoS

Resumindo as escolhas e alternativas:

1. **Kinesis Data Streams** foi escolhido em vez do **MSK** por sua integração mais simples e gerenciamento menos complexo.
2. **AWS Glue** foi escolhido para o processamento batch devido à sua simplicidade e flexibilidade para transformação de dados, em vez de soluções como **EMR**.
3. **Lambda** foi priorizado por ser serverless e facilmente escalável, em vez de **Fargate**, que requer mais configuração de infraestrutura.
4. **DynamoDB** foi preferido por ser mais adequado para dados em tempo real de transações bancárias, enquanto **Aurora** é ideal para cargas transacionais complexas.

Essas ferramentas foram selecionadas visando garantir uma arquitetura simples, escalável e facilmente gerenciável para o MVP, com foco na detecção de fraudes em tempo real e processamento eficiente de grandes volumes de dados históricos.

Criação de Kanban (Tarefas e Responsáveis)

Visando a organização no fluxo de trabalho utilizamos o Trello, que é uma ferramenta da metodologia Kanban que possibilita criar um quadro visual que proporciona uma visão clara do progresso do projeto.

O Trello permite definir tarefas e quebrá-las em etapas menores se for necessário, classificar de acordo com a prioridade, definir prazo de entrega e o membro da equipe responsável por cada tarefa.

O projeto foi dividido em **5 sprints**, refletidos em listas no Trello, com cartões detalhando cada tarefa. Dividimos as Sprints em:

Sprint #1
Estruturar o problema a ser resolvido e a estratégia inicial

- **Tarefas:**
    1. Definição do Caso de Uso
        1.1 Definir mercado e negócio
        1.2 Identificar o problema
        1.3 Determinar público-alvo
        1.4 Avaliar viabilidade técnica
        1.5 Elaborar o pitch
    2. Escolher a base de dados inicial
    3. Desenhar a arquitetura
    4. Selecionar stack de ferramentas

Data de conclusão: 02/12/2024

Sprint #2
Criar o planejamento das tarefas e estruturar as bases de dados

- **Tarefas:**
    1. Criar um Kanban para tarefas e responsabilidades.
    2. Modelar camadas de dados
        2.1 Criar **Star Schema** para a análise
        2.2 Criar **Wide Table** para a análise

Data de conclusão: 04/12/2024

Sprint #3:
Estabelecer o escopo mínimo do produto e organizar o ambiente de trabalho

- **Tarefas:**
    1. Definir o MVP.
    2. Elaborar um planner para as entregas do MVP.
    3. Levantar os requisitos necessários.

4. Preparar o ambiente técnico.

Data de conclusão: 09/12/2024

Sprint #4:
Implementar a infraestrutura necessária e o pipeline de dados

- **Tarefas:**
    1. Configuração e desenvolvimento
       1.1 Criar a **Arquitetura Lambda** em nuvem
       1.2 Definir e documentar a base RAW
       1.3 Prover recursos (especificar tamanho, processador, memória etc.)
       1.4 Ingerir dados brutos (RAW)
       1.5 Limpar e padronizar os dados (camada Silver)
       1.6 Disponibilizar dados para analytics (camada Gold)
    2. Documentar todos os processos

Data de conclusão: 15/12/2024

Sprint #5:
Validar, revisar e Criar apresentação

- **Tarefas:**
    1. Desenvolvimento do ppt
    2. Revisão final

Data de conclusão: 16/12/2024

Modelagem das Camadas de Dados

Efetuamos a modelagem Wide Table aos invés da modelagem Star Schema, pois para nosso caso de uso em detecção de fraude em tempo real, a análise precisa ser realizada rapidamente, e o Wide Table nos oferece a simplicidade e agilidade necessária, por ser uma única tabela que fornece os dados das transações e comportamentos.

Legenda Dos Dados

step - mapeia uma unidade de tempo. Neste caso, 1 passo equivale a 1 hora

type - tipos de pagamento: CASH-IN (dinheiro), CASH-OUT (saque), DEBIT (débito), PAYMENT (pagamento) and TRANSFE (transferência)

amount - valor da transação

nameOrig - cliente que efetuou a transação

oldbalanceOrg - antigo saldo antes da transação

newbalanceOrig - novo saldo após a transação

nameDest - destinatário da transação

oldbalanceDest - destinatário do saldo inicial antes da transação. Observe que não há informações para que iniciam com M (Comerciantes)

newbalanceDest - novo destinatário do saldo após a transação. Observe que não há informações para clientes que iniciam com M (Comerciantes)

isFraud - número 1 é fraude e 0 não fraude

isFlaggedFraud - Controle do modelo que sinaliza possíveis fraudes, 0 não é fraude e 1 sinaliza possível fraude

Definição do MVP (Minimum Viable Product)

O objetivo deste MVP é criar uma versão simplificada da arquitetura de dados na AWS para a **detecção de fraudes, contendo apenas o essencial para testar a ideia, para isso iremos:**

- Ingerir dados de transações financeiras e informações complementares (como por exemplo a cidades) em formato CSV
- Criar camadas de dados organizadas: Bronze, Silver e Gold
- Garantir a limpeza e integração dos dados nas camadas intermediárias (Silver) e transformar em formato parquet, para ter os dados armazenados em formato otimizado
- Ter os dados na camada Gold, unificados e preparados para treinar e alimentar um modelo inicial de detecção de fraudes.

Planner MVP

Etapa	**Atividade**	**Duração**
1. Planejamento	Definir requisitos do projeto, arquitetura inicial e ferramentas a serem utilizadas.	1 dia
2. Configuração Inicial	Configurar o ambiente AWS	
3. Ingestão de Dados	Implementar ingestão de dados para a camada Bronze, Carregando os dados brutos em CSV no S3.	1 dia
4. Processamento de Dados	- Limpeza dos dado e conversão dos arquivos para formato parquet, para inserir na camada Silver. - Transformação e unificação para a camada Gold.	1 dia
6. Detecção de Fraudes	- Criar modelo básico de detecção de fraudes com scikit-learn. - Treinar modelo com dados da camada Gold.	1 dia

8. Testes e Validações	Realizar testes de ponta a ponta para ingestão, processamento, detecção e visualização.	1 dia
9. Documentação e Entrega	Documentar a arquitetura, fluxos de dados e instruções para manutenção.	1 dia

Duração total estimada: 6 dias úteis

Parte prática:

Análise e Design

Configuração

Desenvolvimento

Análise e Design

Configuração

Desenvolvimento

Teste e Validação

Link do Repositório:

https://github.com/jphassel/fraud-detection-project

Bibliografia

STICHLER, J. F.; SCHUMACHER, L. The Gift of Customer Complaints. Marketinh Health Services, v. 23, n. 4, p.14-15, Winter, 2003.

IBM. Detecção de fraude. Disponível em: <https://www.ibm.com/br-pt/topics/fraud-detection>. Acesso em: 5 dez. 2024.