Projeto Integrado I

Título: Detecção de Fraudes em Transações Bancárias

Equipe: Daniel – Danillo – Matheus

BlueShift Academy
2024

SUMÁRIO

1. Introdução

A Detecção de fraudes financeiras é um dos principais desafios na área de segurança bancária e financeira. Com o aumento das transações digitais, torna-se essencial desenvolver modelos que possam identificar padrões fraudulentos em tempo real.

Este projeto visa explorar um conjunto de dados de transações financeiras, realizar uma análise exploratória e construir um modelo de detecção de fraudes. As ferramentas utilizadas neste trabalho incluem SQL Server, Power BI e Python.

2. Desenvolvimento

2.1. Requisitos

Para a execução deste projeto, foram utilizados os seguintes requisitos:

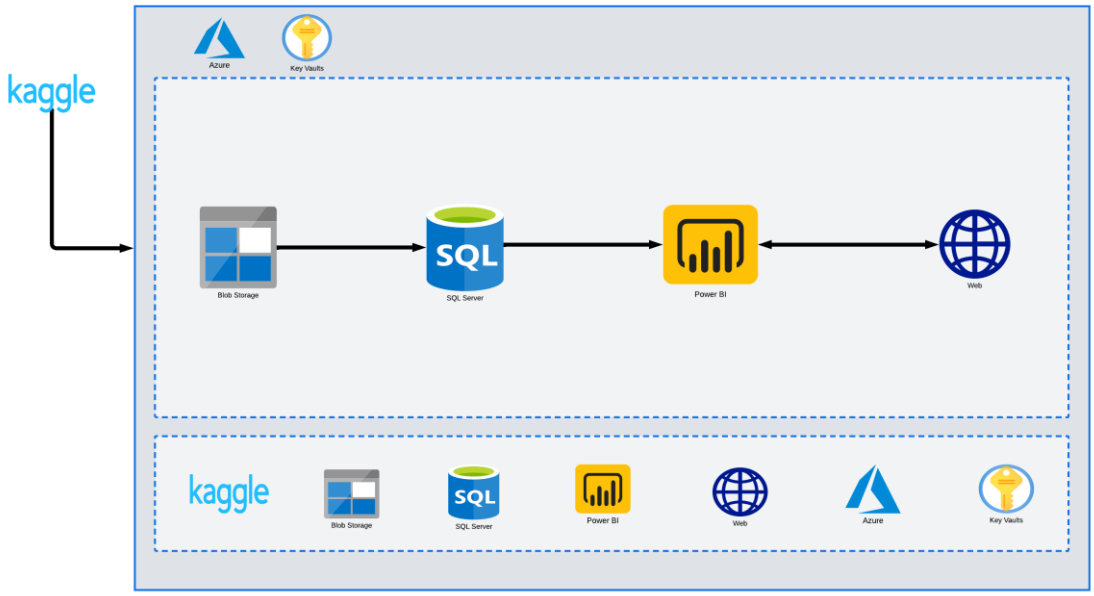
- **Python:** (utilizando bibliotecas como Pandas e Pyodbc) para análise e manipulação de dados;
- **SQL Server:** para armazenamento de dados e realização de consultas;
- **Power BI:** para visualização interativa dos resultados da análise; e
- **Azure Blob Storage e Azure Key Vault:** para gerenciamento e segurança de dados.

- CRUD

- Criar uma funcionalidade que permita listar as últimas 10 transações inseridas no banco, exibindo todas as colunas relevantes;
- Atualização de Valores Fraudulentos. Implementar um recurso para identificar e corrigir valores "estratosféricos" em transações marcadas como fraudulentas, garantindo maior integridade nos dados.
- Criar relatórios que agrupem transações por tipo (Tipo), como saques, transferências, etc., destacando as mais propensas a fraudes.
- Verificar inconsistências entre SaldoAntigoOrigem, SaldoNovoOrigem, SaldoAntigoDestino e SaldoNovoDestino para identificar possíveis falhas no sistema ou comportamentos suspeitos.

- Desenvolver filtros para listar transações acima de um determinado valor, tanto fraudulentas quanto não fraudulentas.
- Permitir consultas detalhadas de todas as transações realizadas por uma conta específica (origem ou destino), incluindo fraudes associadas.

3. Arquitetura Utilizada



- Fonte de Dados (Kaggle): Os dados das transações bancárias são extraídos da plataforma Kaggle. Estes dados incluem informações detalhadas das transações, essenciais para detectar padrões fraudulentos.

- Armazenamento Seguro e Organização (Azure)

- Key Vaults: As credenciais de acesso e chaves de segurança são armazenadas no Azure Key Vault, garantindo segurança no acesso ao ambiente.
- Blob Storage: Os dados brutos são enviados para o Azure Blob Storage, que funciona como um repositório para armazenar arquivos em nuvem.
- Storage Container: Dentro do Blob Storage, é criado um Container para organizar e gerenciar os dados.

- Transformação dos Dados:

Os dados armazenados no Blob Storage são processados utilizando Python/Pandas.

Durante esta etapa, as seguintes ações são realizadas:

- Limpeza: Remoção de inconsistências e valores nulos.
- Transformação: Alteração no formato dos dados para padronizar colunas, normalizar valores e calcular campos necessários para análise.

- Banco de Dados (SQL Server)

Após a transformação, os dados são enviados para um banco de dados SQL hospedado na nuvem.

SQL Server: Os dados são armazenados em tabelas estruturadas, otimizadas para consultas e análises futuras.

- Interface que utiliza funções CRUD (Create, Read, Update, Delete) para:

- Inserir novos dados de transações;
- Consultar dados para análise ou visualização;
- Atualizar informações conforme necessário; e
- Excluir registros redundantes ou inválidos.

6. Análises e Visualização

Análises: Utilizando SQL e Python, análises avançadas são conduzidas diretamente no banco de dados para identificar padrões de fraudes. São gerados relatórios e *insights* baseados em critérios como valores, contas, frequência de transações, etc.

Power BI: As informações analisadas são conectadas ao *Power BI*, onde *Dashboards* interativos são criados para visualizar:

- Transações suspeitas.
- Tendências gerais.
- *KPIs* relevantes.

O *Power BI* apresenta esses resultados diretamente ao cliente (usuário final), possibilitando monitoramento em tempo real.

- Usuário Final

O cliente tem acesso aos *Dashboards* no *Power BI* para tomar decisões informadas e visualizar o status de fraudes detectadas e medidas preventivas.

- Resumo

O processo começa com a coleta de dados do *Kaggle* e passa por etapas de armazenamento seguro, transformação com *Python/Pandas*, inserção e análise no *SQL Server*, além de visualização no *Power BI*. O sistema é capaz de oferecer relatórios dinâmicos e insights precisos ao usuário final, garantindo eficiência na detecção de fraudes.

3.1. Dicionário de Dados

Tabela	projeto_daniel_danillo_matheus			
Descrição	Armazena as informações de transações Bancárias			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de Domínio (Not Null)
Tempo	Número equivalente ao tempo da transação	int		Not Null
Tipo	Tipo de transação	varchar	13	Not Null
Valor	Valor da transação	float		Not Null
ContaOrigem	Conta de onde foi transicionado o valor	varchar	15	Not Null
SaldoAntigoOrigem	Saldo antigo da conta antes da transação do valor	float		
SaldoNovoOrigem	Saldo novo da conta após a transação do valor	float		
ContaDestino	Conta de destino para onde foi transicionado o valor	varchar	15	Not Null
SaldoAntigoDestino	Saldo antigo da conta destinatária, antes da transação do valor	float		
SaldoNovoDestino	Saldo novo da conta destinatária, após a transação do valor	float		
Fraude	Valor de 0 é 'Não Fraudulento' e Valor de 1 é 'Fraudulento'	tinyint		Not Null

4. Objetivos

4.1. Objetivo Geral

O objetivo principal deste projeto é realizar uma análise exploratória e identificar padrões de fraudes em transações financeiras. Além disso, o projeto visa fornecer visualizações interativas que permitam a detecção de comportamentos fraudulentos em grandes volumes de dados.

4.2. Objetivos Específicos

1. Transformação dos dados brutos para um formato adequado para análise.
2. Carregar os dados para o *SQL Server* e realizar consultas exploratórias.
3. Criar visualizações no *Power BI* para identificar padrões de fraudes e transações suspeitas.
4. Identificar os tipos de transações mais suscetíveis a fraudes e outros padrões relevantes.
5. Documentar o processo de análise e fornecer recomendações para detecção de fraudes em sistemas financeiros.

5. Materiais e Métodos

5.1. Descrição das Tecnologias Utilizadas

1. ***Python (Pandas e Pyodbc):***
    - ***Pandas*** foi utilizado para manipulação dos dados, realizando a limpeza e transformação das informações.
    - ***Pyodbc*** foi utilizado para a integração entre Python e SQL Server, permitindo o carregamento dos dados e execução de consultas SQL.
    - ***MatplotLib*** foi utilizado para gerar Gráficos estáticos para auxiliar no entendimento das analises
2. ***SQL Server:***
    - O SQL Server foi utilizado como banco de dados relacional para armazenar e consultar as transações financeiras.
    - Foi fundamental para a realização de consultas SQL e manipulação eficiente dos dados.
3. ***Power BI:***
    - O Power BI foi utilizado para criar dashboards interativos e visualizações que facilitam a interpretação dos dados e dos padrões de fraude encontrados.
    - Ferramentas como gráficos de barras, pizza e linha foram utilizadas para representar visualmente a distribuição de fraudes e o comportamento das transações.
4. ***Azure Blob Storage e Key Vault:***
    - ***Azure Blob Storage*** foi utilizado para armazenar o arquivo original com dados das transações financeiras.
    - ***Azure Key Vault*** foi usado para gerenciar e proteger credenciais e segredos, como as conexões ao *SQL Server* e ao *Blob Storage*, garantindo a segurança dos dados sensíveis.
5. ***Miro*****:**
    - ***Miro*** foi utilizado para o planejamento e colaboração durante o projeto, permitindo que a equipe trabalhasse de forma ágil e eficiente.

      Com o *Miro*, conseguimos visualizar fluxos de trabalho, mapear processos e acompanhar as tarefas de forma interativa, promovendo uma comunicação eficaz durante todas as etapas do projeto

6. ***GitHub:***
    - ***GitHub*** foi utilizado para versionamento de código, controle de alterações e compartilhamento entre os membros da equipe.

O *GitHub* facilitou a colaboração entre os desenvolvedores, garantindo que as modificações no código fossem acompanhadas e integradas de maneira eficiente, além de possibilitar a revisão de código.

6. Conclusão

O projeto de detecção de fraudes financeiras foi bem-sucedido em identificar padrões nas transações financeiras que indicam comportamentos fraudulentos. O uso de *SQL Server*, *Python e Power BI* facilitou a manipulação dos dados e a criação de visualizações interativas, permitindo uma análise eficiente dos dados. Além disso, a integração com o *Azure* garantiu segurança e eficiência no manuseio de grandes volumes de dados.

7. Referências

1. ***Kaggle Dataset: Fraud Detection Dataset.***
    - Link: Dados
2. **Documentação do *Power BI*.**
    - Link: *Power BI*
3. **Documentação do *SQL Server*.**
    - Link: *SQL Server*
4. **Documentação do *Azure*.**
    - Link: *Azure*
5. **Repositório *GitHub***
    - Link: *GitHub*