
Faculdade
XPe
RELATÓRIO
PROJETO
APLICADO
PÓS-GRADUAÇÃO

XP Educação

Relatório do Projeto Aplicado

Detecção de Fraudes em Transações Financeiras

Guilherme Azeredo Mendes

Orientador(a): Luiz Eduardo Labriola

Setembro/2024

Guilherme Azeredo Mendes

XP EDUCAÇÃO

RELATÓRIO DO PROJETO APLICADO

Detecção de Fraudes em Transações Financeiras

Relatório de Projeto Aplicado desenvolvido para fins de conclusão do curso de Pós-Graduação em Ciência de Dados para Mercado Financeiro.

Orientador (a): Luiz Eduardo Labriola

São João da Boa Vista - SP

Setembro/2024

Sumário

1. CANVAS do Projeto Aplicado

Figura conceitual, que representa todas as etapas do Projeto Aplicado.

1.1 Desafio

1.1.1 Análise de Contexto

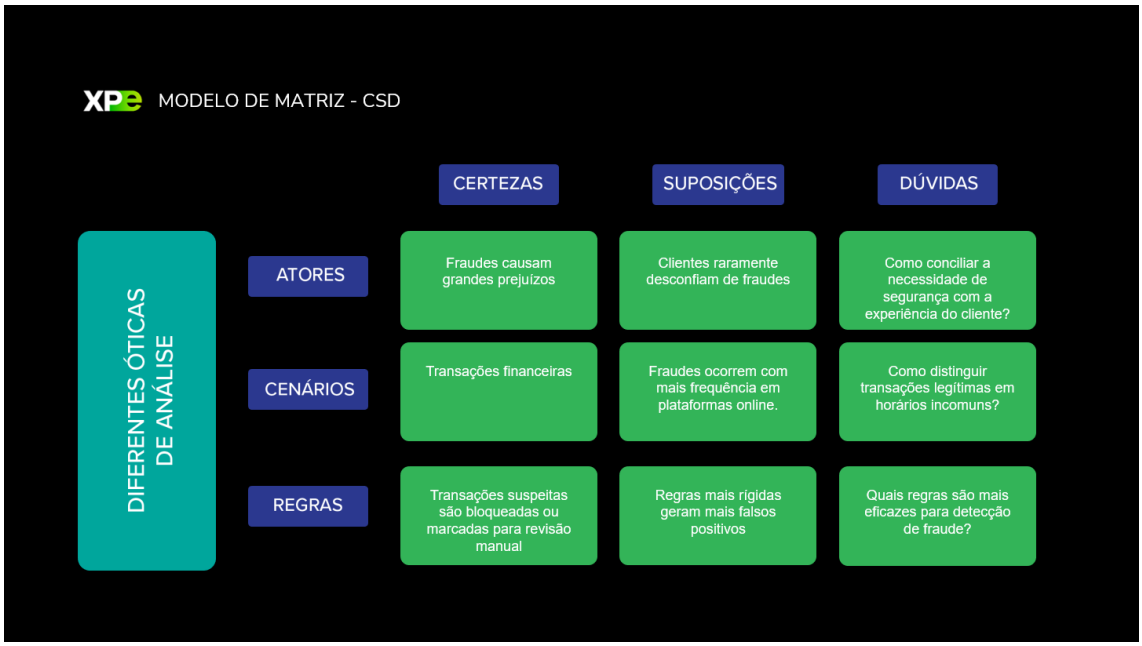
A detecção de fraudes em transações financeiras é uma tarefa crítica no setor financeiro, com o crescimento das compras online e o uso de cartões, o risco de fraudes aumentou, levando a perdas financeiras significativas. Por isso, nos dias atuais são cruciais sistemas que identifiquem rapidamente transações fraudulentas, protegendo tanto os clientes quanto as instituições.

O objetivo desse projeto concentra-se em desenvolver um algoritmo para identificar essas transações fraudulentas de forma eficiente, garantindo que os clientes não sejam penalizados injustamente.

Principais desafios:

- Desbalanceamento dos dados:
    - Na maioria dos casos, os datasets de fraudes financeiras são fortemente desbalanceados, ou seja, apenas uma pequena parte dos dados são de transações fraudulentas. Dessa forma modelos de machine learning podem ser enviesados a classificar todas as transações como legítimas.
- Falsos positivos e negativos:
    - Podem existir muitos falsos positivos (transações legítimas classificadas como fraude) que pode causar insatisfação do cliente e perda de confiança no sistema. Também podem ocorrer falsos negativos (transações fraudulentas não detectadas) que podem causar prejuízos financeiros significativos.
- Histórico de transações:
    - A detecção de fraudes é influenciada por padrões temporais. Transações que ocorrem em horários fora do comportamento usual do cliente podem ser um indicativo de fraude. No entanto, o conjunto de dados pode ser pontual sobre transações individuais, sem contexto adicional como o histórico de transações.
- Evolução dos padrões de fraude:
    - Os fraudadores frequentemente adaptam suas táticas para contornar sistemas de detecção. Isso significa que um modelo treinado em dados históricos pode se tornar obsoleto à medida que novos padrões de fraude aparecem.

Matriz CSD



POEMS:

1.1.2 Personas

Persona 1

- Nome: João Alves
- Idade: 35 anos
- Ocupação: Gerente de Projetos
- **Perfil:** João é um cliente atento com a segurança das suas transações online. Ele faz compras online frequentes, principalmente em plataformas internacionais. No entanto, valoriza a confiança e o bom funcionamento do seu banco e espera que suas transações ocorram sem problemas.
- **Frustração:** Se preocupa em ser vítima de fraude e que suas informações pessoais sejam comprometidas. João já teve transações legítimas bloqueadas e isso o irrita.

Persona 2

- Nome: Fernanda Costa
- Idade 29 anos
- Ocupação: Analista de Fraude em uma Instituição Financeira
- Perfil: Fernanda é uma profissional dedicada, que trabalha revisando transações suspeitas identificadas pelos sistemas automáticos do banco. Sua função é identificar padrões e comportamentos que indiquem fraudes. Ela busca constantemente equilibrar eficiência com precisão, para não gerar estresse aos clientes com bloqueios desnecessários.

- **Frustração:** Fernanda sente que, apesar de bons sistemas, alguns casos são complexos e exigem investigações manuais demoradas.

1.1.3 Benefícios e Justificativas

Um algoritmo de detecção de fraudes em transações financeiras oferece diversos benefícios e justificativas para o seu desenvolvimento. Esses benefícios incluem:

- Redução de prejuízos financeiros:
  - A implementação de um algoritmo de detecção de fraudes ajuda as instituições financeiras a minimizar perdas significativas causadas por transações fraudulentas, protegendo a empresa e seus clientes.
- Melhoria da experiência do cliente:
  - Com a diminuição de falsos positivos garante aos clientes legítimos transações sem interrupções aumentando a satisfação e retenção do cliente.
- Eficiência operacional:
  - Automatizar a identificação de fraudes pode reduzir o volume de trabalho manual para os analistas de fraude, permitindo que se concentrem em casos mais complexos melhorando a precisão da detecção de fraudes da instituição.

Existem diversas justificativas para o desenvolvimento de um algoritmo para detecção de fraudes, dentre elas estão:

- Aumento de número de fraudes:
  - Com o crescimento do comércio digital e das transações online, as fraudes financeiras também aumentaram. Instituições financeiras precisam de sistemas avançados para enfrentar esse desafio.
- Automatização de processos:
  -
- Tecnologias avançadas disponíveis:
  - As tecnologias de machine learning e inteligência artificial superam os métodos tradicionais de detecção baseados em regras estáticas, tendo um grande avanço na detecção de transações fraudulentas.

Proposta de valor

A **solução de detecção de fraudes** oferecida por este projeto utiliza algoritmos avançados de machine learning para identificar padrões anormais em transações financeiras, protegendo as instituições financeiras de prejuízos e assegurando uma experiência fluida e segura para os clientes. Com alta precisão na detecção e redução de falsos positivos, a proposta do projeto permite que as transações legítimas aconteçam sem interrupções, ao mesmo tempo que bloqueia fraudes antes que ocorram, proporcionando confiança e eficiência operacional.

Blueprint

Atores envolvidos:

- **Clientes:** Realizam transações financeiras.
- **Fraudadores:** Tentam realizar transações fraudulentas.
- **Instituições Financeiras:** Provedoras do serviço financeiro e responsáveis pela segurança das transações.
- **Analistas de Fraude:** Revisam e monitoram transações suspeitas.
- **Desenvolvedores e Engenheiros de Dados:** Constroem e mantêm o sistema de detecção de fraudes.

Etapas:

Fase	Ação do Cliente	Ação Interna (Instituição)	Interações Visíveis	Interações Invisíveis
Início da Transação	O cliente realiza uma transação.	Instituição recebe a transação pelo sistema.	Cliente interage com o app ou site bancário.	Sistema de pagamento verifica os dados da transação.
Verificação Inicial	O cliente aguarda a aprovação da transação.	Sistema de detecção de fraudes analisa a transação com base nos dados disponíveis.	Sistema indica que está processando a transação.	Algoritmo de detecção analisa padrões e dados comportamentais.
Análise de Risco	Nenhuma ação do cliente.	Sistema marca a transação como “segura” ou “suspeita”. Se suspeita, encaminha para verificação manual.	Transação pode ser aprovada automaticamente ou ser revisada.	Sistema de priorização classifica transações para análise manual.
Decisão Automática	Cliente pode receber notificação de transação bloqueada ou aprovada.	Se a transação for legítima, o sistema aprova automaticamente. Se suspeita, notifica o cliente.	Notificação, no app ou site, de aprovação ou bloqueio.	Se necessário, transação é enviada para análise manual.
Revisão Manual (Se Necessário)	Nenhuma ação do cliente.	Analistas de fraude revisam manualmente as transações suspeitas e	Cliente pode ser notificado para confirmar a transação.	Analista verifica padrões e contexto adicional para avaliar.

		tomam uma decisão final.		
Conclusão	Cliente recebe a aprovação ou bloqueio final da transação.	O sistema atualiza seus modelos com base nos resultados da transação, melhorando para futuras detecções.	Transação finalizada e cliente notificado do status final.	Modelo de machine learning é treinado com novos dados.

1.1.4 Hipóteses

Hipóteses para direcionar o desenvolvimento do projeto:

- **Hipótese 1: A precisão dos modelos de detecção de fraudes pode ser melhorada com o uso de machine learning**
    - Algoritmos de machine learning podem detectar fraudes com mais precisão do que métodos tradicionais.
- **Hipótese 2: Reduzir o número de falsos positivos melhora a satisfação e retenção de clientes**
    - Ao diminuir o número de transações legítimas bloqueadas, melhora-se a confiança e satisfação dos clientes.
- **Hipótese 3: A atualização contínua dos modelos de machine learning aumenta a eficácia na detecção de novas fraudes**
    - Atualizar regularmente os modelos com novos dados de fraudes permite que o sistema acompanhe as novas técnicas de fraudes.
- Hipótese 4: Comunicação em tempo real reduz erros de bloqueio
    - Enviar uma notificação ao cliente para que ele confirme a autenticidade de uma transação suspeita, ao invés de bloquear automaticamente, pode evitar bloqueios indevidos.

Hipóteses	Observações
Modelos de machine learning melhoram a precisão.	Testar diferentes algoritmos de machine learning e comparar sua precisão com regras tradicionais.
Reduzir falsos positivos melhora a satisfação.	Analisar o feedback e retenção dos clientes antes e após a implementação de melhorias no modelo.
Atualizações contínuas melhoram a eficácia.	Atualizar periodicamente os modelos com novos dados e medir sua eficácia antes e depois das atualizações.
Comunicação em tempo real reduz erros de bloqueio.	Implementar notificações de confirmação para clientes e medir a quantidade de transações desbloqueadas após a confirmação do cliente.

Matriz de priorização de ideias para criação do algoritmo detector de fraudes em transações financeiras

Critérios de avaliação:

- **Impacto no Negócio: Avalia o** quanto a implementação dessa ideia pode melhorar os resultados do negócio, seja na redução de fraudes, aumento de satisfação dos clientes ou redução de custos operacionais.
- **Viabilidade Técnica: Avalia** a dificuldade para implementar essa ideia, considerando as habilidades da equipe e a infraestrutura disponível.
- **Custo: Avalia o** custo financeiro para implementar e manter essa ideia, incluindo recursos de desenvolvimento e infraestrutura.
- **Tempo de Implementação: Avalia o** tempo necessário para implementar a ideia, desde o desenvolvimento até a validação e integração.
- **Urgência: Avalia** a necessidade de implementar essa ideia, seja por pressão regulatória, aumento de fraudes ou necessidades imediatas dos clientes.
- **Facilidade de Escala: Avalia** quanto a solução pode ser expandida à medida que mais dados e transações forem adicionados ao sistema.

Critérios de Avaliação	Pontuação
Impacto no Negócio	8
Viabilidade Técnica	7
Custo	5
Tempo de Implementação	6
Urgência	7
Facilidade de Escala	7
Pontuação Total	40

Com base na análise realizada na matriz, o critério mais forte no projeto é o impacto no negócio, seguido pela facilidade de escala e urgência. O critério com menor pontuação é o custo, o que indica que as funcionalidades propostas podem ser relativamente caras de implementar, mas são justificadas pelo alto impacto esperado no negócio.

A média geral do projeto é 7.14, isso indica um bom equilíbrio entre os critérios, com viabilidade e impacto suficientes para seguir com o desenvolvimento das funcionalidades prioritárias.

1.2 Solução

1.2.1 Objetivo SMART

Objetivo para o projeto de detecção de fraudes em transações financeiras:

- **S (Específico):** Desenvolver um sistema de detecção de fraudes em transações financeiras utilizando algoritmos de machine learning, com o objetivo de facilitar a identificação de fraudes e diminuir o número de falsos positivos, aumentando a satisfação dos clientes.
- **M (Mensurável):** O sucesso será medido com a comparação dos indicadores de fraudes identificadas e falsos positivos antes e depois da implementação do projeto.
- **A (Atingível):** A meta pode ser atingida com o uso de modelos de machine learning juntamente com o treinamento contínuo dos modelos para se ajustarem a novos padrões de fraudes.
- **R (Relevante):** A redução de fraudes não identificadas e falsos positivos é crucial para o mercado financeiro, melhorando a confiança dos clientes e reduzindo as perdas financeiras.
- **T (Temporal):** O objetivo é desenvolver o projeto dentro de 6 meses, podendo ter um tempo de implementação de 6 meses por instituição financeira.

Portanto, o objetivo SMART para o projeto de detecção de fraudes em transações financeiras é:
Desenvolver um projeto base que possa ser vendido e implementado para diferentes instituições financeiras para melhorar a identificação de fraudes, diminuir o número de falsos positivos e, consequentemente, aumentar a satisfação de seus clientes.

1.2.2 Premissas e Restrições

Premissas e restrições são aspectos importantes a serem considerados no projeto de detecção de fraudes em transações financeiras. A seguir serão respectivamente apresentadas.

Premissas:

- **Disponibilidade de Dados:** Os dados históricos de transações financeiras, incluindo registros de fraudes e transações legítimas, devem estar disponíveis para alimentar os modelos de machine learning.
- **Qualidade dos Dados:** Os dados fornecidos serão limpos e estruturados, permitindo uma análise eficiente e precisa.
- **Recursos Técnicos adequados:** Disponibilidade de infraestrutura de TI necessária para suportar o desenvolvimento, implementação e operação do sistema.
- **Conformidade com Regulações:** O sistema será implantado dentro dos padrões regulatórios aplicáveis ao mercado financeiro, principalmente em relação à segurança de dados como LGPD.
- **Engajamento da Equipe:** A implementação do sistema irá necessitar do envolvimento de diversos setores da instituição financeira.
- **Treinamento Contínuo do Modelo:** A equipe de ciência de dados será devidamente treinada para a atualização contínua dos modelos de dados, permitindo a detecção de novos padrões de fraude.

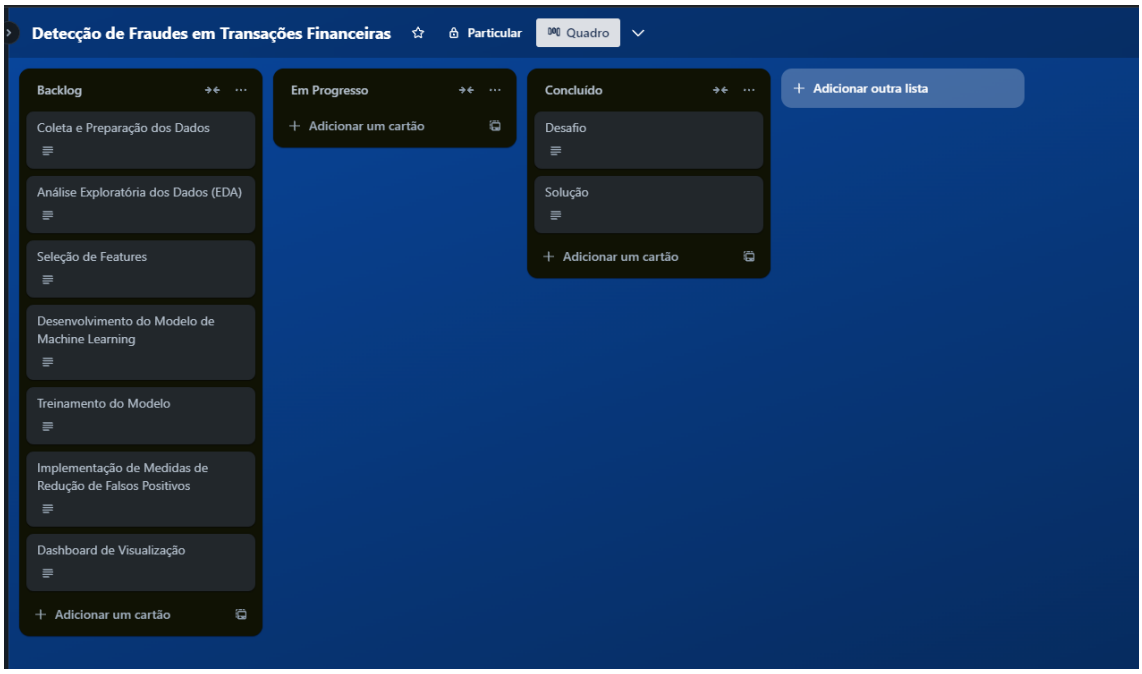
Restrições:

- **Orçamento Limitado:** O projeto deve ser desenvolvido dentro do orçamento aprovado.
- **Prazo de Implementação:** O sistema de detecção de fraudes em transações financeiras deve estar implementado e funcional em um prazo máximo de 6 meses.
- **Capacidade Computacional:** A infraestrutura pode ter limitações de processamento para lidar com grandes volumes de dados em tempo real.
- **Interoperabilidade com Sistemas Existentes:** O sistema de detecção de fraudes deve ser compatível com os sistemas de pagamento já existentes na instituição financeira.
- **Disponibilidade de Profissionais Especializados:** A instituição financeira pode ter limitações no número de profissionais com experiência em machine learning e ciência de dados, podendo restringir o ritmo da implementação.
- **Políticas de Privacidade e Segurança:** As abordagens de coleta e processamento de dados devem estar em conformidade com as políticas de privacidade, restringindo a captura de informações pessoais além do necessário para o funcionamento do sistema.

Matriz de Riscos:

Risco	Probabilidade	Impacto	Nível de Risco
Qualidade dos Dados Insuficientes ou Inadequado.	Alta	Alta	Crítico
Falta de Conformidade com Regulamentações.	Média	Alta	Alto
Sobrecarga de Infraestrutura de TI.	Média	Alta	Alto
Falta de Engajamento da Equipe.	Média	Média	Médio
Limitações no Orçamento.	Alta	Alta	Crítico
Atrasos no Cronograma de Implementação.	Média	Alta	Alto
Baixa Precisão do Modelo de Machine Learning.	Média	Alta	Alto
Falta de Especialistas em Ciência de Dados.	Alta	Média	Alto
Falsos Positivos em Volume Elevado.	Média	Alta	Alto
Baixa Aceitação pelo Cliente.	Baixa	Média	Médio

1.2.3 Backlog de Produto

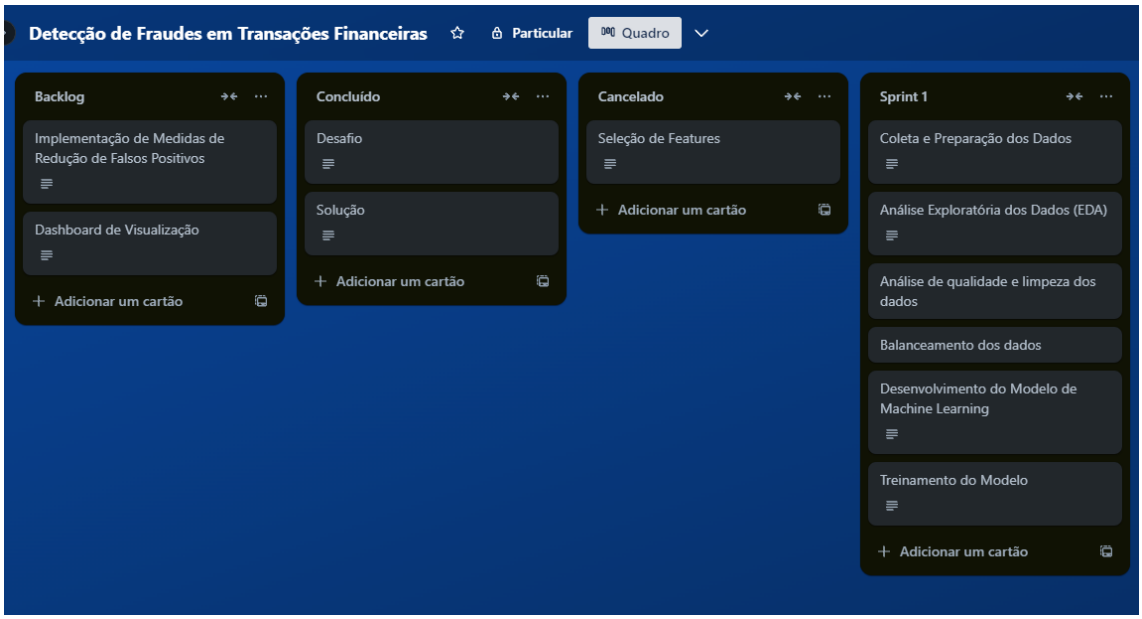


2. Área de Experimentação

2.1 Sprint 1

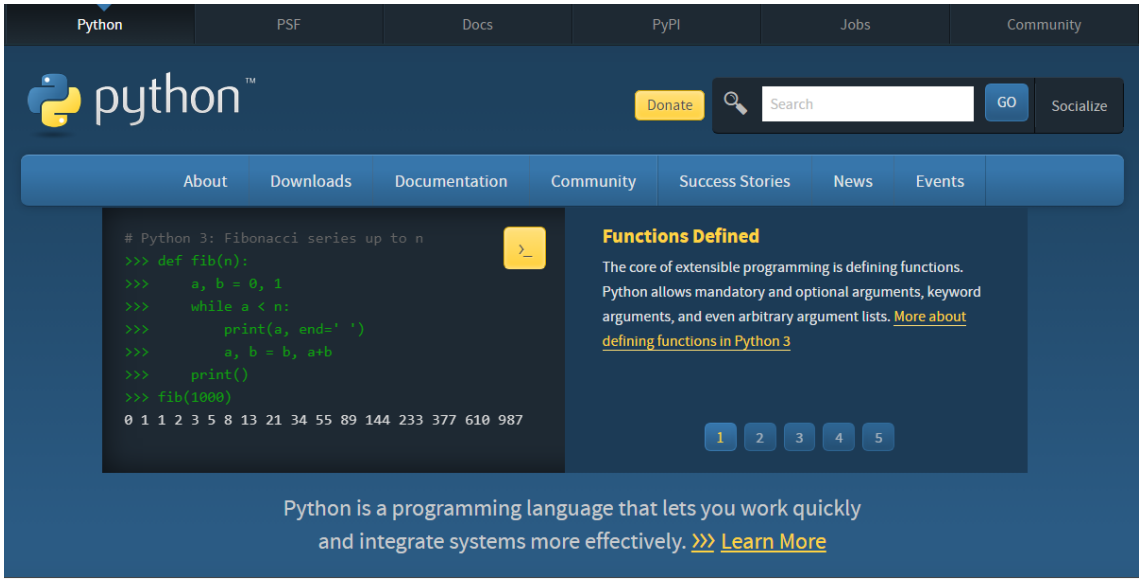
2.1.1 Solução

- Evidência do planejamento:



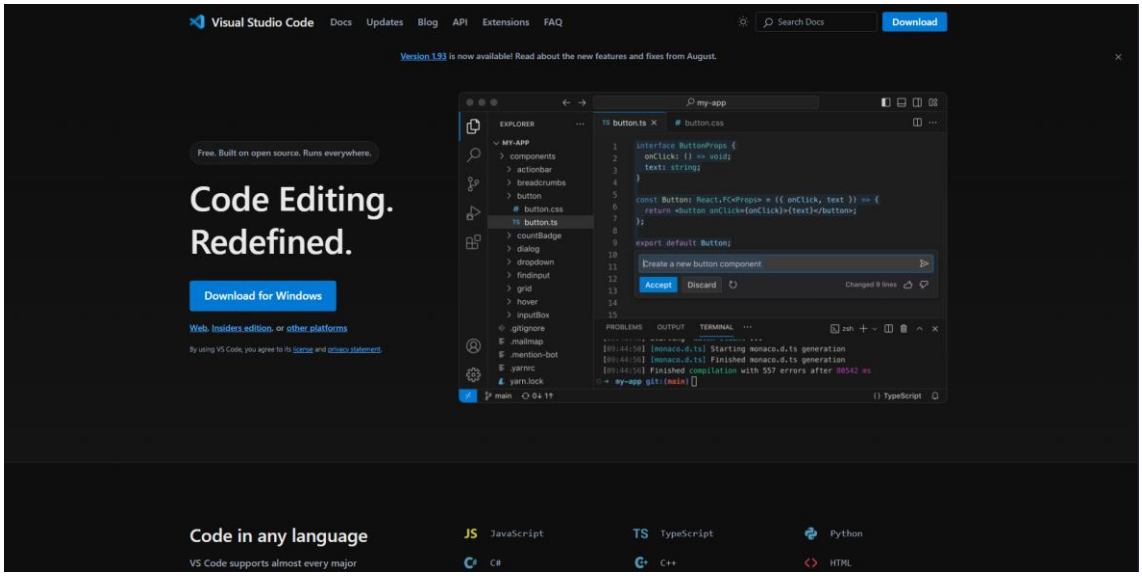
- Evidência da execução de cada requisito:

Seleção da linguagem: Python

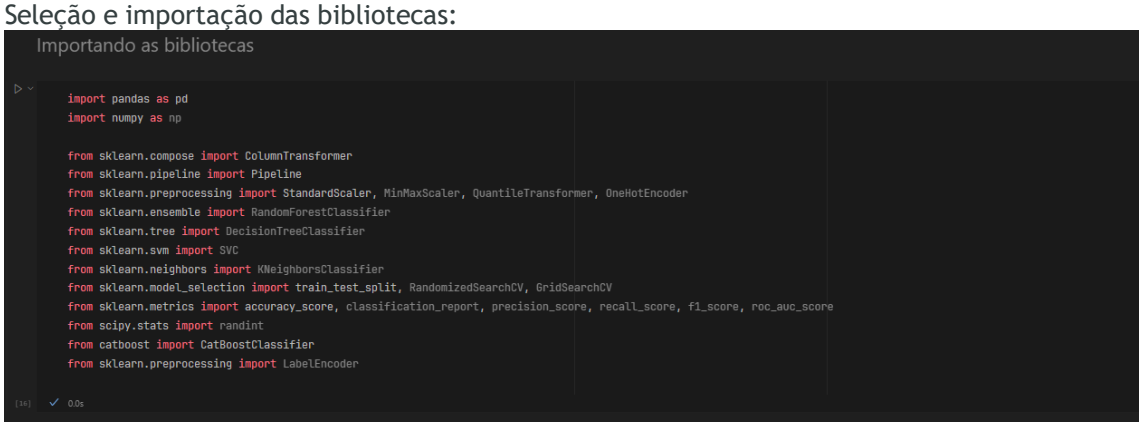


Fonte: https://www.python.org/

Seleção da plataforma de programação: Visual Studio Code

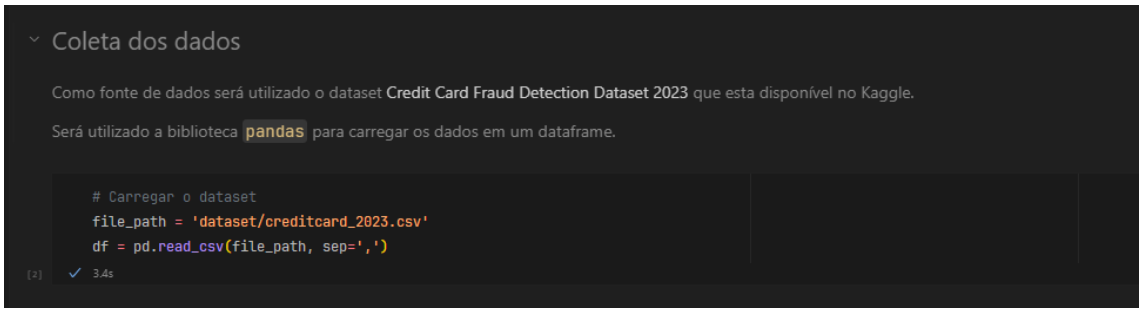


Seleção e importação das bibliotecas:

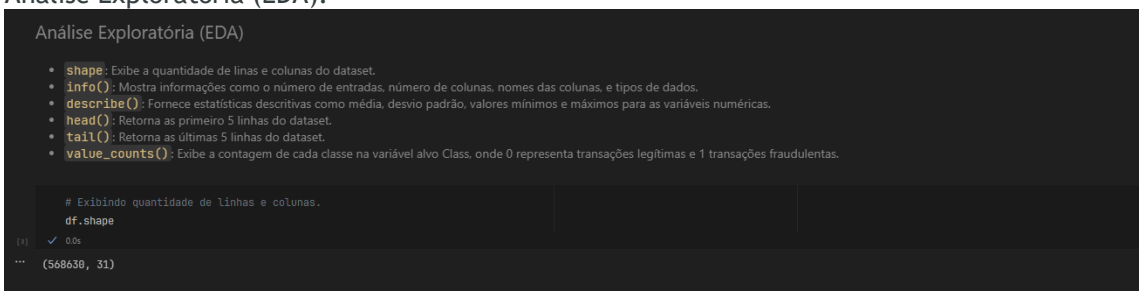


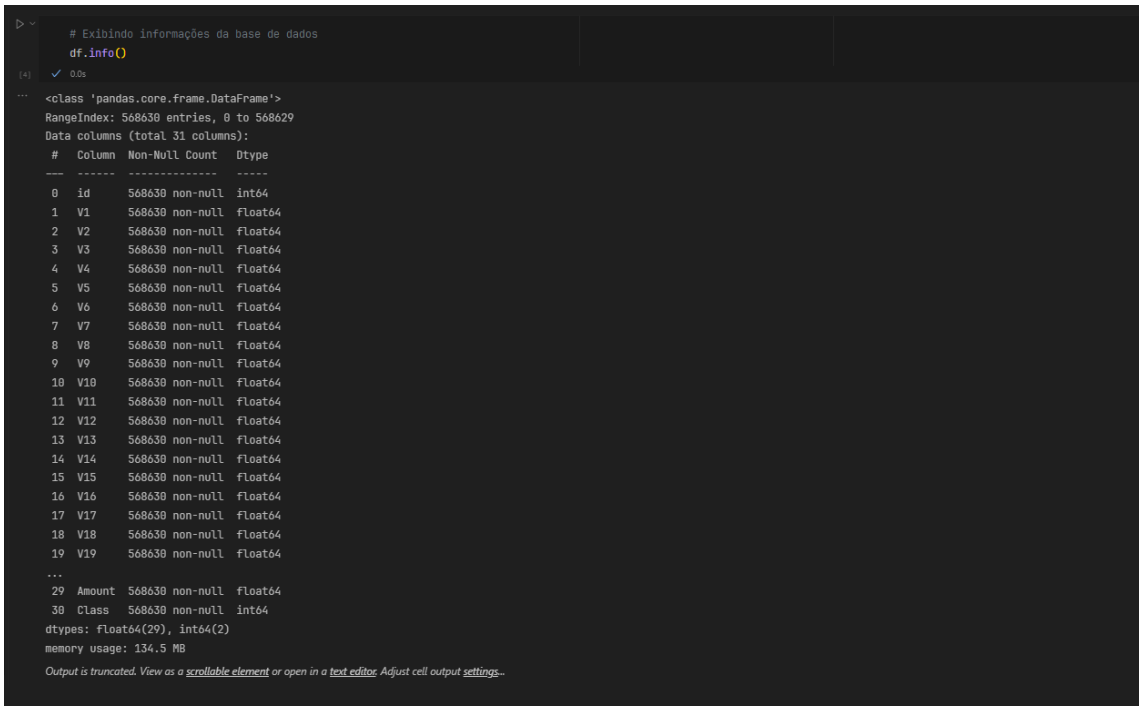
- Evidência dos resultados:

Coleta dos dados:



Análise Exploratória (EDA):





```
# Informações quantitativas do dataset
df.describe()
```

	id	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26
count	568630.000000	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	...	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05
mean	284314.500000	-5.638058e-17	-1.319545e-16	-3.518788e-17	-2.879008e-17	7.997245e-18	-3.958636e-17	-3.198898e-17	2.109273e-17	3.998623e-17	...	4.758361e-17	3.948640e-18	6.194741e-18	-2.799036e-18	-3.178905e-17	-7.497417e-18
std	164149.486122	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	...	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00
min	0.000000	-3.495584e+00	-4.996657e+01	-3.183760e+00	-4.951222e+00	-9.952786e+00	-2.111111e+01	-4.351839e+00	-1.075634e+01	-3.751919e+00	...	-1.938252e+01	-7.734798e+00	-3.029545e+01	-4.067968e+00	-1.361263e+01	-8.226969e+00
25%	142157.250000	-5.652859e-01	-4.866777e-01	-6.492987e-01	-6.560203e-01	-2.934955e-01	-4.458712e-01	-2.835329e-01	-1.922572e-01	-5.687446e-01	...	-1.664408e-01	-4.904892e-01	-2.376289e-01	-6.515801e-01	-5.541485e-01	-6.318948e-01
50%	284314.500000	-9.363846e-02	-1.358939e-01	3.528579e-04	-7.376152e-02	8.108788e-02	7.871758e-02	2.333659e-01	-1.145242e-01	9.252647e-02	...	-3.743065e-02	-2.732881e-02	-5.968903e-02	1.590123e-02	-8.193162e-03	-1.189208e-02
75%	426471.750000	8.326582e-01	3.435552e-01	6.285380e-01	7.070047e-01	4.397368e-01	4.977881e-01	5.259548e-01	4.729905e-02	5.592621e-01	...	1.479787e-01	4.638817e-01	1.557153e-01	7.007374e-01	5.500147e-01	6.728879e-01
max	568629.000000	2.229046e+00	4.361865e+00	1.412583e+01	3.201536e+00	4.271689e+01	2.616840e+01	2.178730e+02	5.958040e+00	2.027006e+01	...	8.087080e+00	1.263251e+01	3.170763e+01	1.296564e+01	1.462151e+01	5.623285e+00

8 rows × 31 columns

```
# Visualizando as primeiras linhas do dataset
df.head()
```

	id	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0	-0.260648	-0.469648	2.496266	-0.083724	0.129681	0.732898	0.519014	-0.130006	0.727159	...	-0.110552	0.217606	-0.134794	0.165959	0.126280	-0.434824	-0.081230	-0.151045	17982.10	0
1	1	0.985100	-0.356045	0.558056	-0.429654	0.277140	0.428605	0.406466	-0.133118	0.347452	...	-0.194936	-0.605761	0.079469	-0.577395	0.190090	0.296503	-0.248052	-0.064512	6531.37	0
2	2	-0.260272	-0.949385	1.728538	-0.457986	0.074062	1.419481	0.743511	-0.095576	-0.261297	...	-0.005020	0.702906	0.945045	-1.154666	-0.605564	-0.312895	-0.300258	-0.244718	2513.54	0
3	3	-0.152152	-0.508959	1.746840	-1.090178	0.249486	1.143312	0.518269	-0.065130	-0.205698	...	-0.146927	-0.038212	-0.214048	-1.893131	1.003963	-0.515950	-0.165316	0.048424	5384.44	0
4	4	-0.206820	-0.165280	1.527053	-0.448293	0.106125	0.530549	0.658849	-0.212660	1.049921	...	-0.106984	0.729727	-0.161666	0.312561	-0.414116	1.071126	0.023712	0.419117	14278.97	0

5 rows × 31 columns

```
# Visualozando as ultimas linhas do dataset
df.tail()
```

	id	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
568625	568625	-0.833437	0.061886	-0.899794	0.904227	-1.002401	0.481454	-0.370393	0.189694	-0.938153	...	0.167503	0.419731	1.288249	-0.900861	0.560661	-0.006018	3.308968	0.081564	4394.16	1
568626	568626	-0.670459	-0.202896	-0.068129	-0.267328	-0.133660	0.237148	-0.016935	-0.147733	0.483894	...	0.031874	0.388161	-0.154257	-0.846452	-0.153443	1.961398	-1.528642	1.704306	4653.40	1
568627	568627	-0.311997	-0.004095	0.137526	-0.035893	-0.042291	0.121098	-0.070958	-0.019997	-0.122048	...	0.140788	0.536523	-0.211100	-0.448909	0.540073	-0.755836	-0.487540	-0.268741	23572.85	1
568628	568628	0.636871	-0.516970	-0.300889	-0.144480	0.131042	-0.294148	0.580568	-0.207723	0.893527	...	-0.060381	-0.195609	-0.175488	-0.554643	-0.099669	-1.434931	-0.159269	-0.076251	10160.83	1
568629	568629	-0.795144	0.433236	-0.649140	0.374732	-0.244976	-0.603493	-0.347613	-0.340814	0.253971	...	0.534853	-0.291514	0.157303	0.931030	-0.349423	-1.090974	-1.575113	0.722936	21493.92	1

5 rows × 31 columns

```
# Verificar o número de transações fraudulentas e não fraudulentas
df['Class'].value_counts()
```

```
Class
0    284315
1    284315
Name: count, dtype: int64
```

Análise da qualidade e limpeza dos dados:

Análise da qualidade e limpeza dos dados

Nessa etapa serão verificados e tratados os valores ausentes, duplicados e outliers.

Tratamento de dados duplicados

```
# Exibindo os primeiros registros dos dados duplicados no dataset - Não temos
df[df.duplicated()].head()
```

	id	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class

0 rows × 31 columns

Tratamento de dados ausentes

```
# Verifica se existem dados ausentes - Não temos
df.isnull().sum()
```

```
id        0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
```

Balanceamento dos dados:

Balanceamento de dados

```
# Pegando dados de análise e removendo coluna que indica o resultado a ser buscado.
X_dados = df.drop('Class', axis=1)
X_dados
```

0.1s

Python

	id	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount
0	0	-0.260648	-0.469648	2.496266	-0.083724	0.129681	0.732898	0.519014	-0.130006	0.727159	...	0.091202	-0.110552	0.217606	-0.134794	0.165959	0.126280	-0.434824	-0.081230	-0.151045	17982.10
1	1	0.985100	-0.356045	0.558056	-0.429654	0.277140	0.428605	0.406466	-0.133118	0.347452	...	-0.233984	-0.194936	-0.605761	0.079469	-0.577395	0.190090	0.296503	-0.248052	-0.064512	6531.37
2	2	-0.260272	-0.949385	1.728538	-0.457986	0.074062	1.419481	0.743511	-0.095576	-0.261297	...	0.361652	-0.005020	0.702906	0.945045	-1.154666	-0.605564	-0.312895	-0.300258	-0.244718	2513.54
3	3	-0.152152	-0.508959	1.746840	-1.090178	0.249486	1.143312	0.518269	-0.065130	-0.205698	...	-0.378223	-0.146927	-0.038212	-0.214048	-1.893131	1.003963	-0.515950	-0.165316	0.048424	5384.44
4	4	-0.206820	-0.165280	1.527053	-0.448293	0.106125	0.530549	0.658849	-0.212660	1.049921	...	0.247237	-0.106984	0.729727	-0.161666	0.312561	-0.414116	1.071126	0.023712	0.419117	14278.97
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
568625	568625	-0.833437	0.061886	-0.899794	0.904227	-1.002401	0.481454	-0.370393	0.189694	-0.938153	...	-0.751011	0.167503	0.419731	1.288249	-0.900861	0.560661	-0.006018	3.308968	0.081564	4394.16
568626	568626	-0.670459	-0.202896	-0.068129	-0.267328	-0.133660	0.237148	-0.016935	-0.147733	0.483894	...	-0.550260	0.031874	0.388161	-0.154257	-0.846452	-0.153443	1.961398	-1.528642	1.704306	4653.40
568627	568627	-0.311997	-0.004095	0.137526	-0.035893	-0.042291	0.121098	-0.070958	-0.019997	-0.122048	...	-0.076417	0.140788	0.536523	-0.211100	-0.448909	0.540073	-0.755836	-0.487540	-0.268741	23572.85
568628	568628	0.636871	-0.516970	-0.300889	-0.144480	0.131042	-0.294148	0.580568	-0.207723	0.893527	...	0.288186	-0.060381	-0.195609	-0.175488	-0.554643	-0.099669	-1.434931	-0.159269	-0.076251	10160.83
568629	568629	-0.795144	0.433236	-0.649140	0.374732	-0.244976	-0.603493	-0.347613	-0.340814	0.253971	...	-0.621378	0.534853	-0.291514	0.157303	0.931030	-0.349423	-1.090974	-1.575113	0.722936	21493.92

568630 rows × 30 columns

```
# Separando coluna com resultado esperado para comparação a análise do algoritmo.
y_dados = df['Class']
y_dados
```

[14] ✓ 0.0s

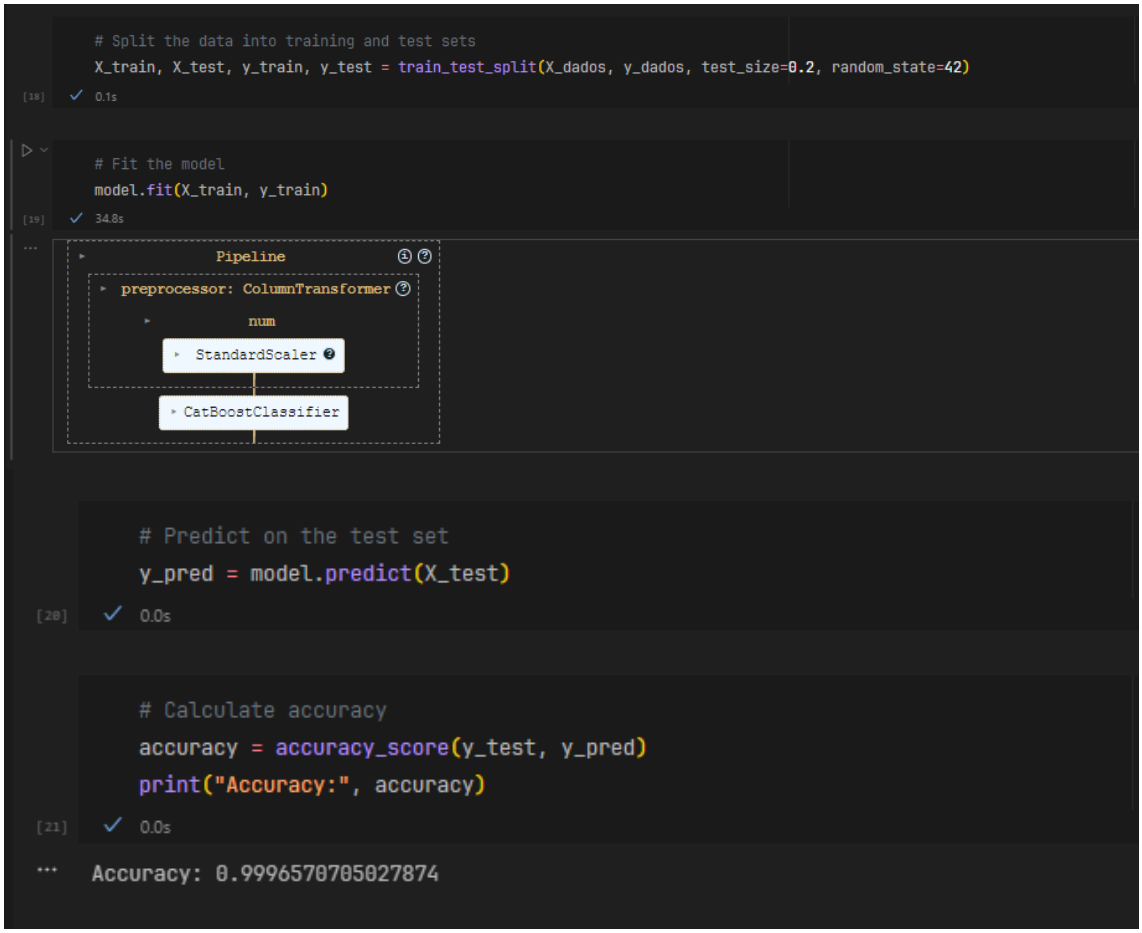
```
0         0
1         0
2         0
3         0
4         0
         ..
568625    1
568626    1
568627    1
568628    1
568629    1
Name: Class, Length: 568630, dtype: int64
```

Desenvolvimento do modelo de machine learning:

```
# Define numeric features (remove categorical columns)
numeric_features = X_dados.select_dtypes(include=['int64', 'float64']).columns.tolist()

# Define preprocessing steps
numeric_transformer = Pipeline(steps=[
    ('scaler', StandardScaler())])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features)])

# Define the model
model = Pipeline(steps=[('preprocessor', preprocessor),
                        ('classifier', CatBoostClassifier(verbose=False))])
```

[17] ✓ 0.0s

Treinamento do modelo:



2.1.2 Lições Aprendidas

Foi utilizada a linguagem Python, que possui vasta documentação e ampla comunidade de apoio, sendo também uma das linguagens com excelente curva de aprendizagem.

O *Visual Studio Code* (VSCode) é uma das *Integrated Development Environment* (IDE) mais populares entre desenvolvedores por várias razões, como por exemplo, leveza e performance, extensibilidade e plugins, suporte a múltiplas linguagens, ferramentas de desenvolvimento integradas, entre muitas outras.

O dataset escolhido foi o 'Credit Card Fraud Detection Datase 2023', encontrado no site www.kaggle.com. Esse dataset conta com um grande número de dados que foram anonimizados para proteção dos dados.

As bibliotecas escolhidas são essenciais para o desenvolvimento de um algoritmo de machine learning em python, desde a coleta dos dados, análise dos mesmo e até a criação e treinamento do modelo de machine learning.

A etapa de análise exploratória (EDA) foi fundamental para entender melhor o conjunto de dados.

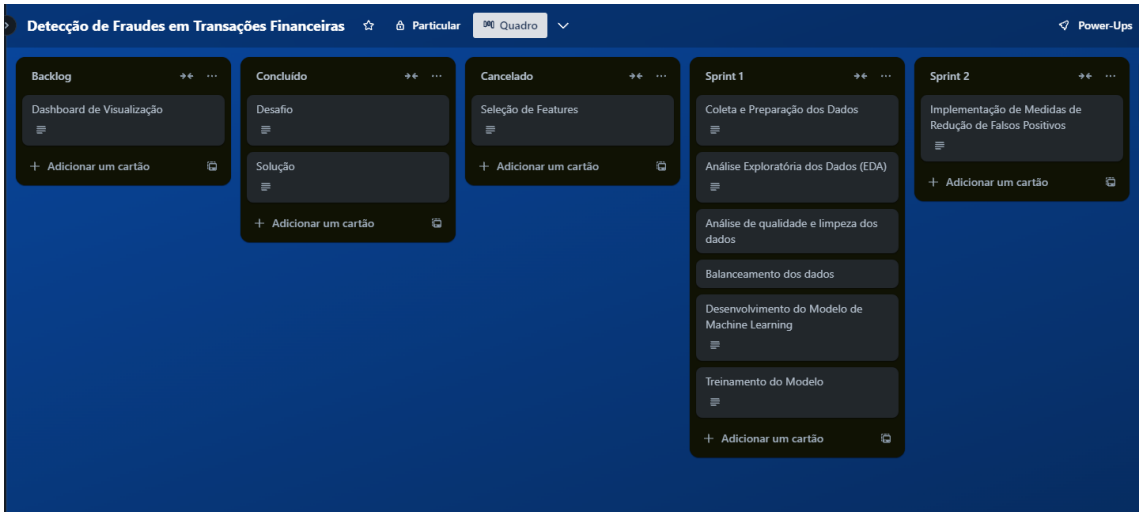
Foram realizadas algumas alterações no planejamento que foram entendidas como não necessárias ou novas tarefas entendidas como necessárias.

- Seleção de Features: Essa tarefa foi entendida como não necessária para o desenvolvimento desse projeto, pois trata-se de um dataset público com dados anonimizados o que não possibilita a identificação de cada coluna.
- Análise de qualidade e limpeza dos dados: Essa etapa é muito importante para o desenvolvimento de um modelo de machine learning pois melhora e eficácia do modelo e evita resultados errados. Não haviam dados ausentes nem duplicados nesse dataset.
- Balanceamento dos dados: Essa também é uma etapa crucial no desenvolvimento de modelos de machine learning pois um conjunto de dados desbalanceado pode prejudicar o desempenho e resultado do modelo

2.2 Sprint 2

2.2.1 Solução

- Evidência do planejamento:



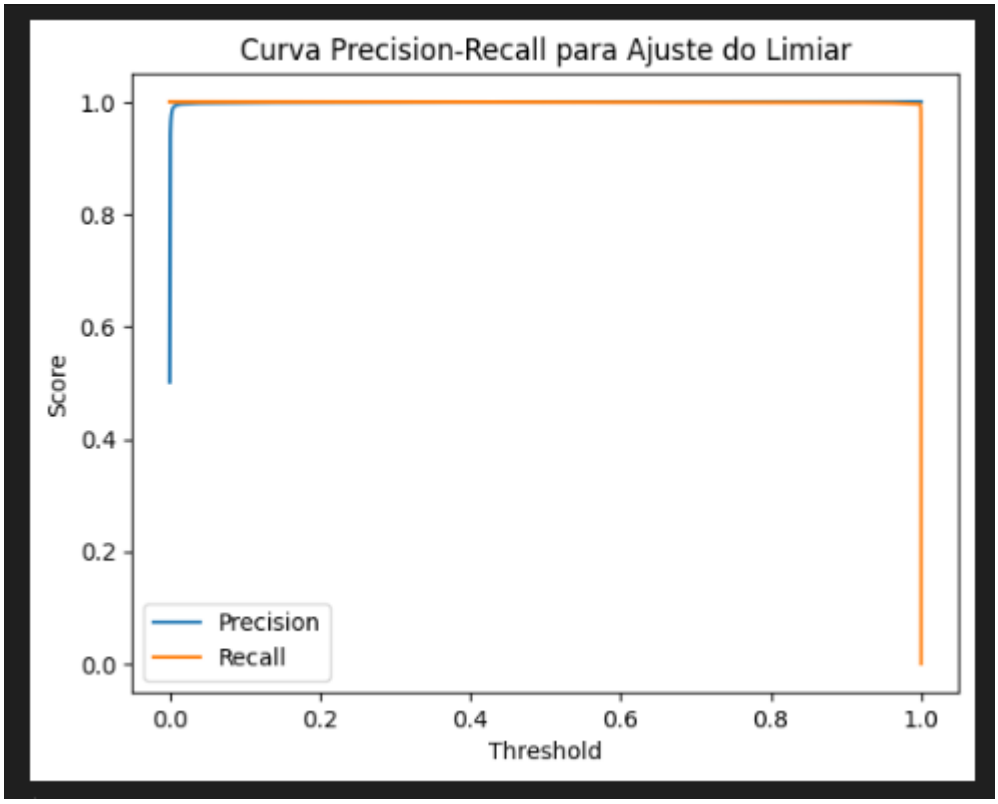
- Evidência da execução de cada requisito:

```
plt.plot(thresholds, precisions[:-1], label="Precision")
plt.plot(thresholds, recalls[:-1], label="Recall")
plt.xlabel("Threshold")
plt.ylabel("Score")
plt.title("Curva Precision-Recall para Ajuste do Limiar")
plt.legend()
plt.show()
```

✓ 0.4s

- Evidência dos resultados:

```
Accuracy com limiar ajustado: 0.9996570705027874
AUC-ROC Score: 0.9999701192680868
Relatório de Classificação:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56750
           1       1.00      1.00      1.00     56976

    accuracy                           1.00    113726
   macro avg       1.00      1.00      1.00    113726
weighted avg       1.00      1.00      1.00    113726
```



2.2.2 Lições Aprendidas

Com base nos resultados obtidos na Sprint 2 pode-se concluir que o modelo está altamente eficaz na detecção de fraudes e, ao mesmo tempo, minimiza falsos positivos. O ajuste do limiar foi uma medida eficaz para garantir que o modelo fosse mais conservador, classificando uma transação como fraudulenta apenas se a probabilidade fosse alta o suficiente, aumentando a precisão sem comprometer o recall. A acurácia alta, o AUC-ROC próximo de 1, e o F1-score perfeito indicam que o modelo está fazendo um excelente trabalho em identificar corretamente as transações fraudulentas e legítimas.

- AUC-ROC

É uma métrica de avaliação utilizada para medir o desempenho de modelos de classificação binária que combina dois conceitos

- **ROC** (Receiver Operating Characteristic): É uma curva que mostra a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos em diferentes limiares de decisão do modelo. A curva ROC ajuda a avaliar como o modelo equilibra esses dois aspectos à medida que o limiar muda.
- **AUC** (Area Under the Curve): Se refere a área sob a curva ROC. O valor do AUC representa a probabilidade de que o classificador ordene corretamente uma amostra positiva (fraude) à frente de uma negativa (transação legítima). O valor de AUC varia entre 0 e 1:
    - AUC = 1: Indica um modelo perfeito, que classifica corretamente todas as amostras.
    - AUC = 0.5: Indica um modelo que faz previsões aleatórias, sem poder de discriminação entre as classes.
    - AUC < 0.5: Indica que o modelo está fazendo previsões piores do que o aleatório, classificando erradamente.

- F1-Score

É uma métrica de desempenho utilizada em problemas de classificação, especialmente em casos onde os dados são desbalanceados, ou seja, uma classe é muito mais frequente que a outra. Ele é a média harmônica entre precisão e recall, proporcionando uma única métrica que equilibra ambas as medidas.

- **Precisão** (Precision): Mede a proporção de exemplos que foram classificados como positivos (fraudes) e que realmente são positivos. Ou seja, quantas das previsões de fraudes estão corretas.

$$Precisão = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positovos + Falsos\ Positivos}$$

- **Recall** (Sensibilidade): Mede a proporção de exemplos positivos que foram corretamente identificados pelo modelo. Ou seja, quantas fraudes reais foram detectadas pelo modelo.

$$Recall = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Negativos}$$

O F1-Score combina essas duas métricas em uma única métrica, proporcionando um equilíbrio entre elas:

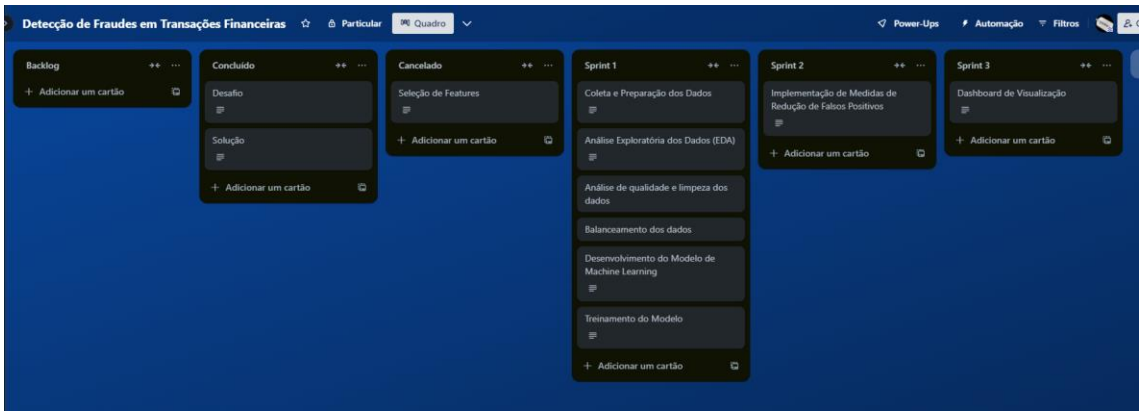
$$F1 = 2 * \frac{Precisão * Recall}{Precisão + Recall}$$

- **F1-Score = 1**: Significa que o modelo tem tanto uma precisão quanto recall perfeitos.
- **F1-Score = 0**: Indica que o modelo falhou completamente em capturar corretamente os exemplos positivos.

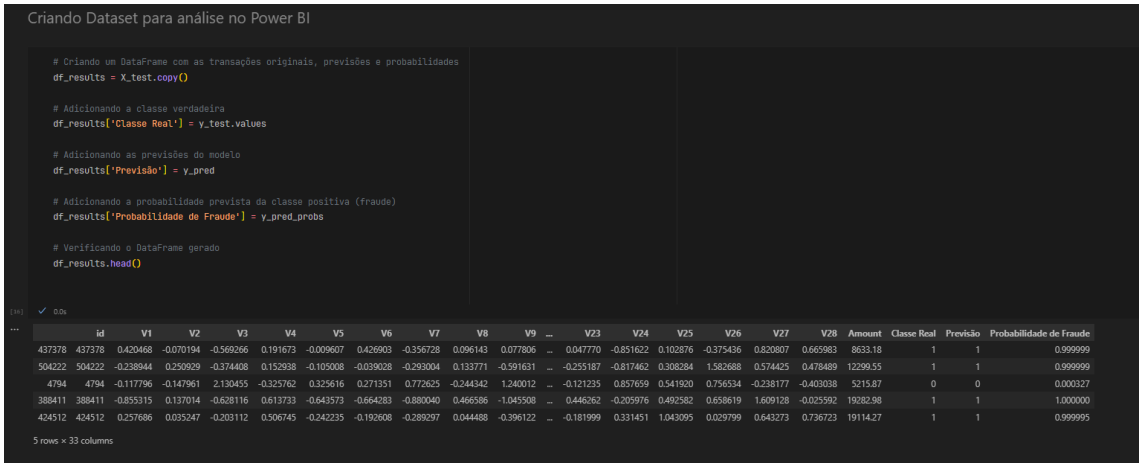
2.3 Sprint 3

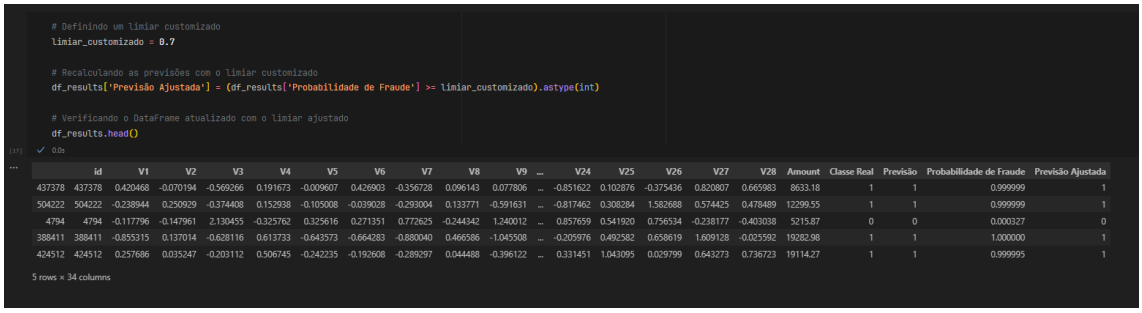
2.3.1 Solução

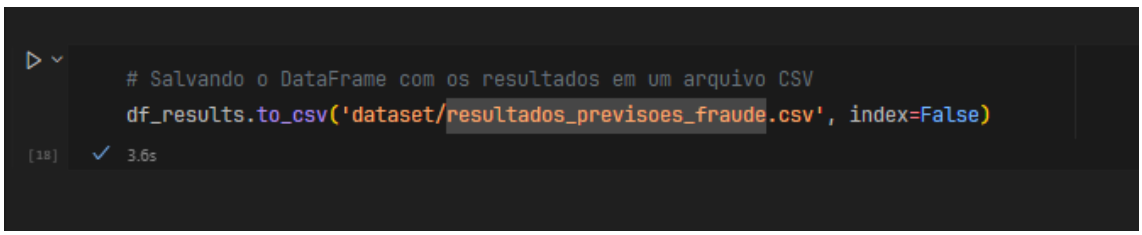
- Evidência do planejamento:



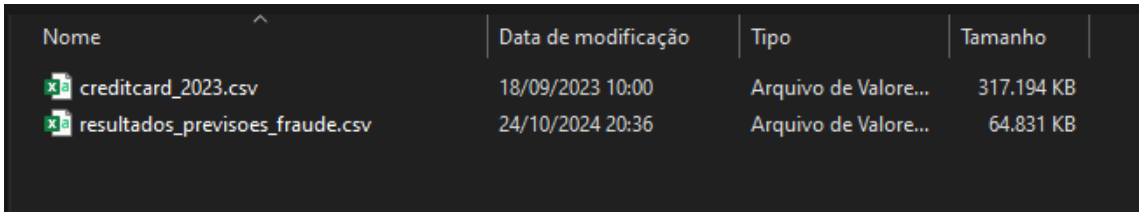
- Evidência da execução de cada requisito:

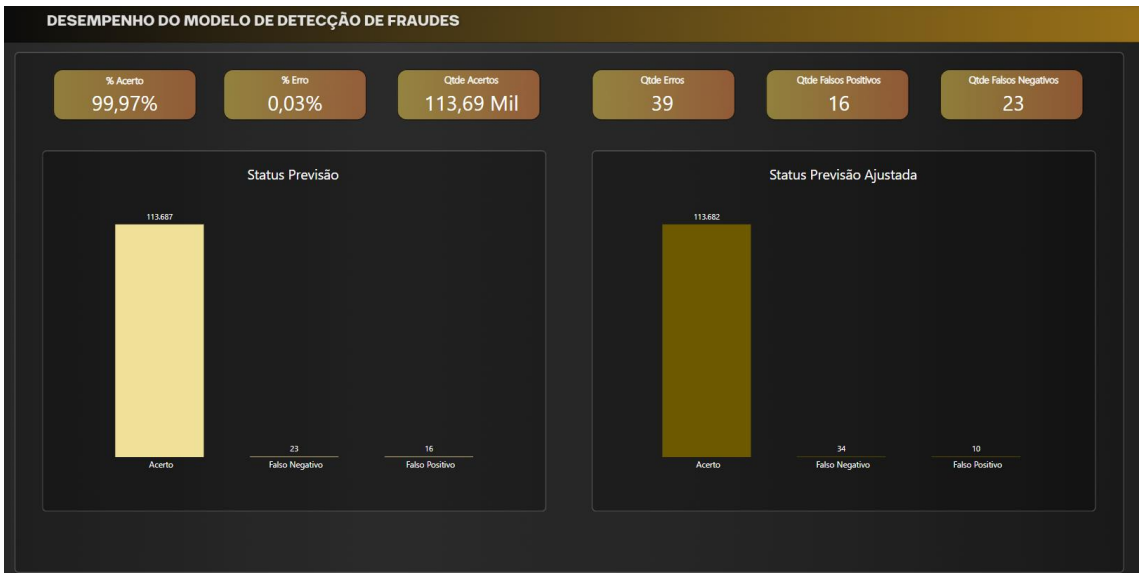






- Evidência dos resultados:





2.3.2 Lições Aprendidas

Durante o processo de integrar os resultados gerados em Python com o Power BI, percebemos como é importante estruturar bem os dados. A criação do dataset com previsões ajustadas e probabilidades foi essencial para permitir análises detalhadas e ricas no Power BI.

O Power BI se mostrou uma ferramenta eficaz para explorar os resultados do modelo, especialmente usando gráficos de barras e matrizes de confusão, que ajudaram a identificar onde o modelo precisava de melhorias, como em falsos positivos e negativos. Estruturar corretamente as colunas do dataset (Classe Real, Previsão, Probabilidade de Fraude, Previsão Ajustada) foi crucial para que as visualizações fluíssem sem problemas.

Ao criar o dashboard, ficou evidente que gráficos simples e diretos são ideais para comunicar os resultados a um público não técnico, facilitando a compreensão da performance do modelo. Além disso, o dashboard ofereceu uma forma eficiente de monitorar o desempenho do modelo ao longo do tempo, permitindo ajustes rápidos e melhorias contínuas.

3. Considerações Finais

3.1 Resultados

O projeto alcançou resultados sólidos e promissores na detecção de fraudes financeiras. O modelo desenvolvido apresentou alta precisão e, após ajustes, conseguiu reduzir significativamente os falsos positivos, o que reflete a eficácia da abordagem adotada. Esse avanço foi fundamental para garantir a confiabilidade do modelo, além de reduzir os alarmes indevidos, algo essencial para aplicações em ambientes financeiros sensíveis.

Entre os pontos positivos, destacamos a possibilidade de monitoramento em tempo real através do dashboard em Power BI, que facilitou a comunicação dos resultados com todas as partes envolvidas, permitindo uma visão completa e integrada do desempenho do modelo. Os dados estruturados de maneira apropriada no Power BI contribuíram para uma experiência mais intuitiva e acessível, especialmente para aqueles menos familiarizados com aspectos técnicos.

Por outro lado, algumas dificuldades foram encontradas, como a necessidade de ajustes constantes no limiar do modelo para manter o equilíbrio entre detecção e precisão. Outra dificuldade foi a manipulação de um volume de dados elevado, que exigiu recursos computacionais avançados e otimizações. Apesar dos desafios, as experiências vivenciadas durante o processo proporcionaram um aprendizado profundo sobre o uso de machine learning na detecção de fraudes e o valor de uma comunicação clara por meio de visualizações.

3.2 Contribuições

O projeto trouxe inovações importantes para o contexto de detecção de fraudes, integrando machine learning com visualizações interativas e dinâmicas. Em comparação a métodos tradicionais de monitoramento, o modelo apresentou a vantagem de adaptação rápida a novos dados, além de uma precisão aprimorada devido ao treinamento supervisionado. A visualização em Power BI tornou os dados acessíveis a diferentes stakeholders, oferecendo insights em tempo real e permitindo uma tomada de decisão mais ágil.

Essa integração entre o modelo de machine learning e o Power BI representa uma contribuição valiosa, permitindo que o processo de detecção de fraudes seja mais transparente e informativo. Ao monitorar o desempenho do modelo em tempo real, com métricas relevantes e de fácil interpretação, a solução apresentou melhorias claras em relação a processos tradicionais de detecção.

3.3 Próximos passos

Para aprimorar ainda mais a solução, os próximos passos envolvem a reavaliação contínua do modelo e a incorporação de novos dados, de modo que ele se mantenha atualizado com os padrões de fraude emergentes. Além disso, a integração com um banco de dados e a implementação de um sistema de automação para alertas em tempo real permitirá que as fraudes sejam detectadas com maior rapidez e efetividade.

Outra evolução prevista é a expansão do dashboard com mais indicadores, como tendências de fraude ao longo do tempo e alertas de mudanças no comportamento. Esse aprimoramento possibilitará que o modelo acompanhe padrões sazonais e ofereça uma análise detalhada das fraudes, contribuindo ainda mais para a segurança financeira e para a tomada de decisões assertivas.