

UNIVERSIDADE DE BRASÍLIA
Faculdade do Gama

Aprendizado de Máquina

Mini Trabalho 3

Exploração dos dados

Grupo - 1

Ana Clara Barbosa Borges
André Emanuel Bispo da Silva
Artur Handow Krauspenhar
Gabriel Moura dos Santos
João Artur Leles Ferreira Pinheiro
João Pedro Anacleto Ferreira Machado

Brasília, DF

2025

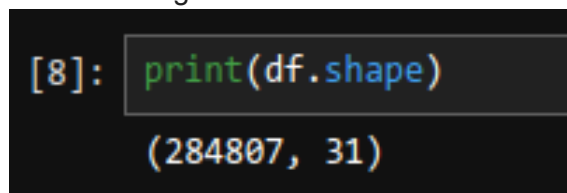
Análise exploratória:

Para obtermos uma compreensão inicial do dataset de transações financeiras, realizamos uma análise exploratória utilizando as bibliotecas *pandas* e *matplotlib* em *Python*. Os passos dessa análise e as visualizações geradas estão detalhados nos notebooks *preliminar.ipynb* e *analise.ipynb*, disponíveis juntamente com este documento.

O dataset contém 284807 linhas e 31 colunas, sendo elas: 'Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20', 'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount' e 'Class'. É possível notar que várias delas possuem nomes iniciados em V, elas representam as colunas anonimizadas devido à segurança e privacidade dos dados.

Iniciamos a exploração verificando a dimensão do dataset, o que nos forneceu o número total de transações (linhas) e as características (colunas). Como ilustra a **Figura 1**, gerada pelo código no notebook *preliminar.ipynb* (Célula 8), o dataset é composto por 284.807 transações e 31 colunas.

Figura 1 - Contagem de linhas e colunas do dataset



```
[8]: print(df.shape)

(284807, 31)
```

Fonte: André Silva, 2025

Para entender a estrutura e os tipos de dados de cada coluna, utilizamos a função `info()` do *pandas*. A **Figura 2**, cuja saída é mostrada abaixo (ver *preliminar.ipynb*, Célula 11), revela que o dataset possui 30 colunas do tipo `float64` e uma coluna (`Class`) do tipo `int64`, e que não há valores nulos em nenhuma das colunas, eliminando a necessidade de tratamento de dados ausentes.

Figura 2 - Avaliação de tipos de dados das colunas

```
[11]: print(df.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column      Non-Null Count  Dtype  
---  --
 0   Time        284807 non-null  float64
 1   V1          284807 non-null  float64
 2   V2          284807 non-null  float64
 3   V3          284807 non-null  float64
 4   V4          284807 non-null  float64
 5   V5          284807 non-null  float64
 6   V6          284807 non-null  float64
 7   V7          284807 non-null  float64
 8   V8          284807 non-null  float64
 9   V9          284807 non-null  float64
10  V10         284807 non-null  float64
11  V11         284807 non-null  float64
12  V12         284807 non-null  float64
13  V13         284807 non-null  float64
14  V14         284807 non-null  float64
15  V15         284807 non-null  float64
16  V16         284807 non-null  float64
17  V17         284807 non-null  float64
18  V18         284807 non-null  float64
19  V19         284807 non-null  float64
20  V20         284807 non-null  float64
21  V21         284807 non-null  float64
22  V22         284807 non-null  float64
23  V23         284807 non-null  float64
24  V24         284807 non-null  float64
25  V25         284807 non-null  float64
26  V26         284807 non-null  float64
27  V27         284807 non-null  float64
28  V28         284807 non-null  float64
29  Amount      284807 non-null  float64
30  Class       284807 non-null  int64   
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
None
```

Fonte: André Silva, 2025

Durante a análise exploratória, identificamos a presença de linhas duplicadas no dataset. A **Figura 3** (saída da Célula 6 em preliminar.ipynb) lista algumas dessas linhas. Foram encontradas um total de 1854 linhas duplicadas. Apesar da duplicação, considerando o domínio de aplicação de detecção de fraudes financeiras, optamos por manter essas linhas por considerar que transações duplicadas podem ser um padrão ou uma anomalia relevante para a construção do modelo.

Figura 3 - Listagem de linhas repetidas do dataset

```
[6]: df[df.duplicated(keep=False)]
```

[6]:		Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	
	32	26.0	-0.529912	0.873892	1.347247	0.145457	0.414209	0.100223	0.711206	0.176066	-0.286717	...	0.046949	0.208105	-0.185548	(
	33	26.0	-0.529912	0.873892	1.347247	0.145457	0.414209	0.100223	0.711206	0.176066	-0.286717	...	0.046949	0.208105	-0.185548	(
	34	26.0	-0.535388	0.865268	1.351076	0.147575	0.433680	0.086983	0.693039	0.179742	-0.285642	...	0.049526	0.206537	-0.187108	0
	35	26.0	-0.535388	0.865268	1.351076	0.147575	0.433680	0.086983	0.693039	0.179742	-0.285642	...	0.049526	0.206537	-0.187108	0
	112	74.0	1.038370	0.127486	0.184456	1.109950	0.441699	0.945283	-0.036715	0.350995	0.118950	...	0.102520	0.605089	0.023092	-0

	283485	171627.0	-1.457978	1.378203	0.811515	-0.603760	-0.711883	-0.471672	-0.282535	0.880654	0.052808	...	0.284205	0.949659	-0.216949	0
	284190	172233.0	-2.667936	3.160505	-3.355984	1.007845	-0.377397	-0.109730	-0.667233	2.309700	-1.639306	...	0.391483	0.266536	-0.079853	-0
	284191	172233.0	-2.667936	3.160505	-3.355984	1.007845	-0.377397	-0.109730	-0.667233	2.309700	-1.639306	...	0.391483	0.266536	-0.079853	-0
	284192	172233.0	-2.691642	3.123168	-3.339407	1.017018	-0.293095	-0.167054	-0.745886	2.325616	-1.634651	...	0.402639	0.259746	-0.086606	-C
	284193	172233.0	-2.691642	3.123168	-3.339407	1.017018	-0.293095	-0.167054	-0.745886	2.325616	-1.634651	...	0.402639	0.259746	-0.086606	-C

1854 rows x 31 columns

Fonte: André Silva, 2025

Além disso foi feita uma análise numérica breve - ver **Figura 4**, gerada a partir da Célula 13 em preliminar.ipynb - sobre a coluna Amount e foi descoberto que o valor mínimo de transação foi de €0.00, tal transação vazia também pode ser sinal de alguma sorte de anomalia, mas como não tivemos acesso à um expert é de difícil discernimento. Dito isso, tais dados ainda podem ser relevantes para o treino do modelo em termos de detecção de anomalias.

Figura 4 - Descrição básica da coluna "Amount"

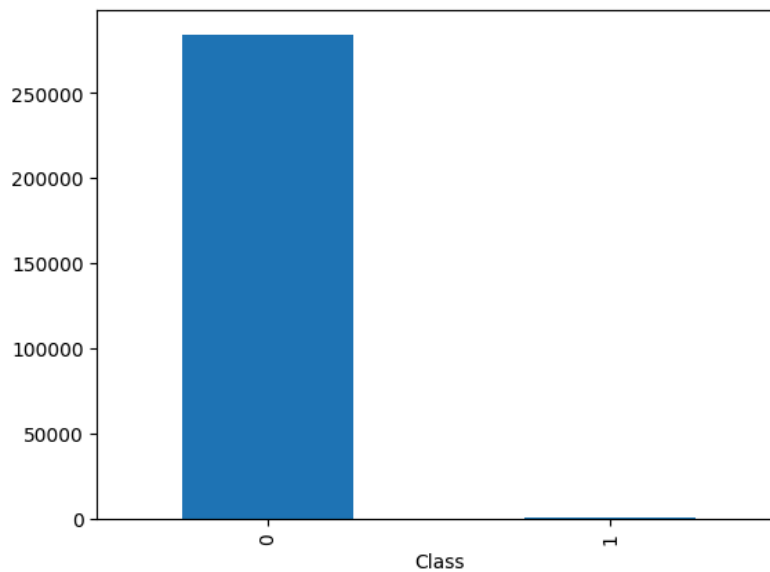
```
[13]: df.describe()['Amount']
```

[13]:	count	284807.000000
	mean	88.349619
	std	250.120109
	min	0.000000
	25%	5.600000
	50%	22.000000
	75%	77.165000
	max	25691.160000
	Name: Amount, dtype: float64	

Fonte: André Silva, 2025

A coluna 'Class' apresenta os valores 0 ou 1, indicando Transações não fraudulentas e Transações fraudulentas, respectivamente. 284315 linhas correspondem a Transações não fraudulentas enquanto 492 representam Transações fraudulentas.

Figura 5 - Gráfico de quantidade por classe



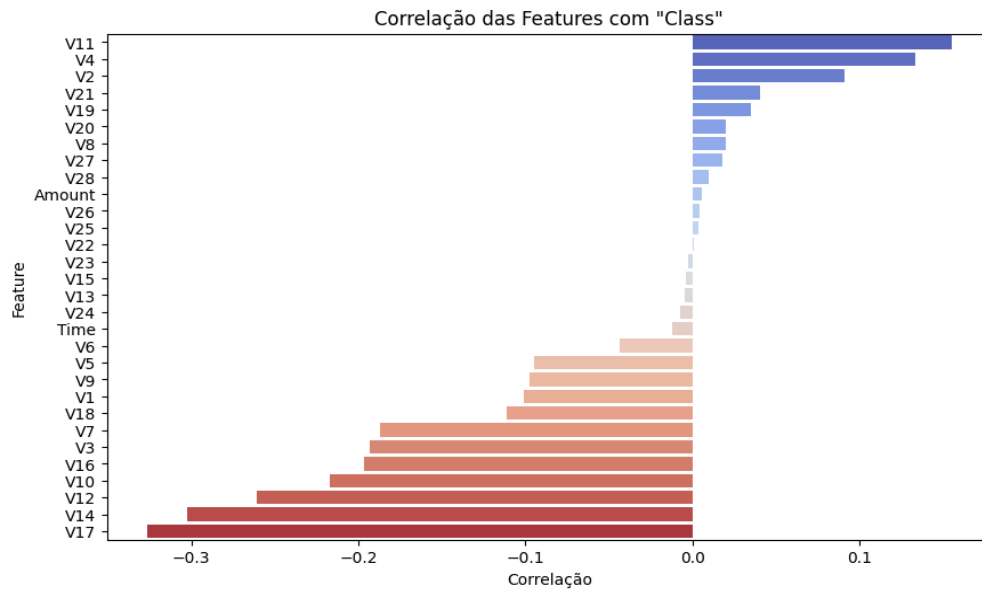
Fonte: Ana Clara Borges, 2025

As correlações e os padrões são apresentados na seção a seguir.

Identificação de padrões e correlações:

O primeiro gráfico apresenta as correlações entre as variáveis do *dataset* e a variável *Class*, que indica se a transação é fraudulenta (1) ou não (0). Utilizando uma barra horizontal colorida, o gráfico mostra a correlação entre as variáveis.

Figura 6 - Gráfico de correlação

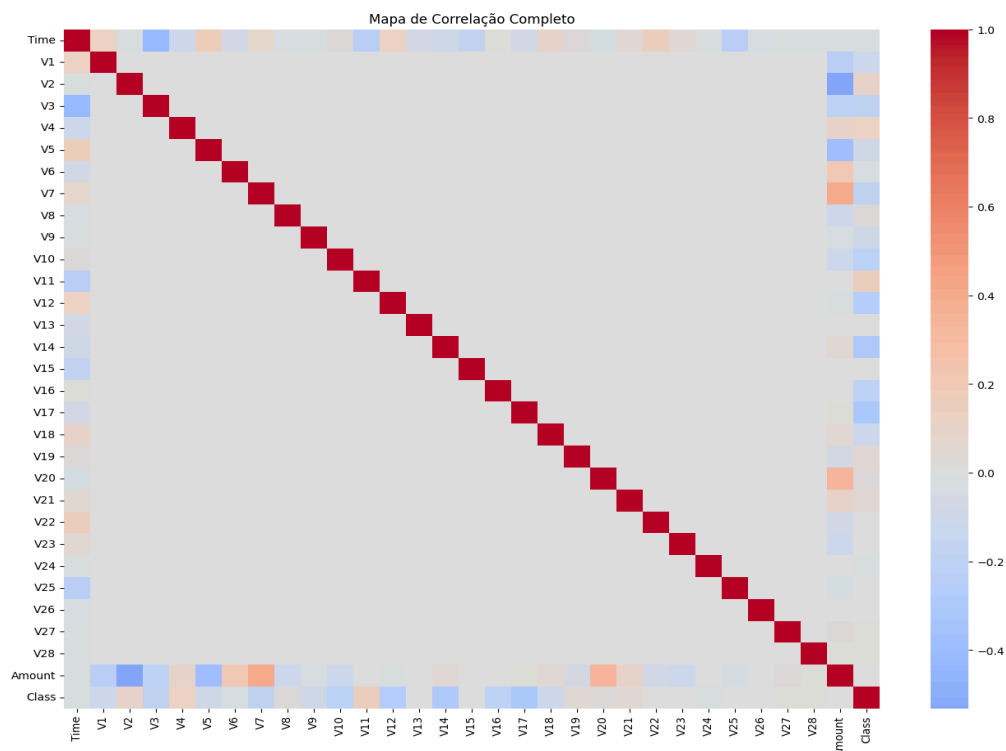


Fonte: Artur Krauspenhar e João Artur Leles, 2025

O gráfico permite avaliar quais features podem ser mais importantes para um modelo de classificação. Segundo o gráfico gerado, as variáveis V14 e V17 seriam as principais features para o modelo, destacando-se também as variáveis V10, V11 e V12.

Além disso, foi criado um heatmap para representar a correlação entre as variáveis, permitindo verificar se existe redundância entre elas. Foi identificado que há uma possível relação de redundância entre V2 e amount.

Figura 7 - Heatmap

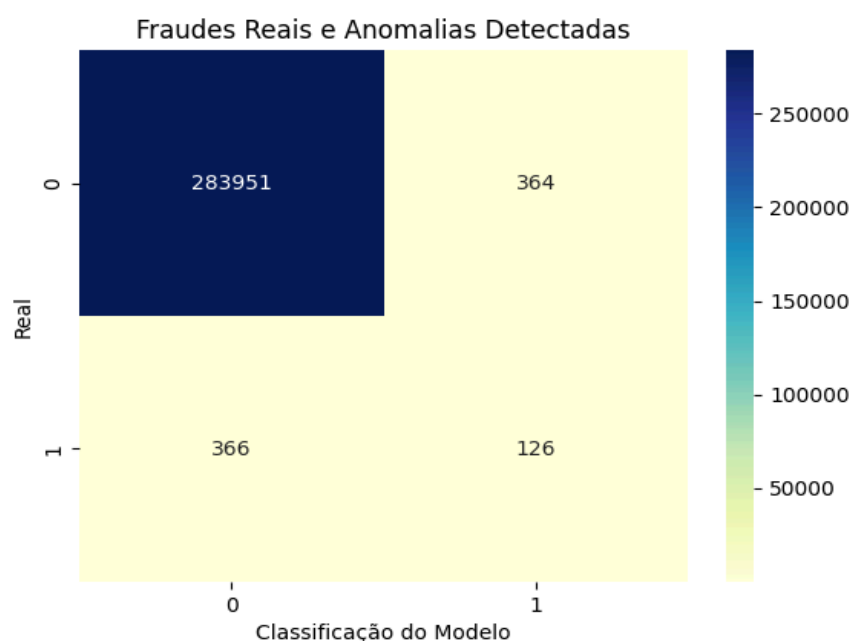


Fonte: Artur Krauspenhar e João Artur Leles, 2025

O *Isolation Forest* é um algoritmo não supervisionado que, quando aplicado a dados de transações financeiras, pode identificar comportamentos atípicos, como transações com valores extremamente altos e padrões de compra fora do comum, os quais podem representar fraudes. (MAKLIN, 2022)]

Diante disso, foi realizado um comparativo entre o modelo criado com o Isolation Forest e a classificação real das transações, utilizando uma matriz de confusão.

Figura 8 - Matriz de confusão



Fonte: Artur Krauspenhar e João Artur Leles, 2025

A matriz de confusão (**Figura 8**) revela que, embora o modelo *Isolation Forest* tenha detectado algumas anomalias que correspondem a fraudes reais (**126 Verdadeiros Positivos**), um número significativo de fraudes reais não foi detectado (**366 Falsos Negativos**). Além disso, o modelo identificou um número considerável de transações normais como anomalias (**364 Falsos Positivos**). Este resultado demonstra a limitação do *Isolation Forest* para este dataset desbalanceado e sugere a necessidade de testar outros algoritmos mais adequados para a detecção de fraudes, como *Random Forest*, *XGBoost* ou modelos supervisionados que lidem com desbalanceamento de classes como oversampling.

Qualidade das visualizações de dados

O **boxplot**, com escala logarítmica, destacou fraudes que costumam envolver valores mais variados e, às vezes, mais altos. O gráfico de **fraudes ao longo do tempo** revelou horários com maior incidência de fraudes, como às 2h e 11h da manhã. O heatmap de correlação possibilitou identificar variáveis mais relevantes, como **V14** e **V17**, e uma possível redundância entre as variáveis **V2** e **Amount**. A matriz de confusão mostrou que o modelo *Isolation Forest* detectou apenas parte das fraudes, indicando a necessidade de testar outros algoritmos mais adequados para dados desbalanceados, como **Random Forest**, **XGBoost** ou modelos supervisionados como oversampling.

De forma geral, os gráficos foram bem construídos, com títulos claros, eixos nomeados e foco na leitura simples, possibilitando uma melhor análise do dataset.

Referência bibliográfica

MAKLIN, Cory. *Isolation Forest*. Medium, 15 jul. 2022. Disponível em: <https://medium.com/@corymaklin/isolation-forest-799fceacdda4>. Acesso em: 23 abr. 2025.