



**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE UNIDADE
ACADÊMICA ESPECIALIZADA EM CIÊNCIAS AGRÁRIAS
ESCOLA AGRÍCOLA DE JUNDIAÍ**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

Gustavo Souza

Relatório da análise de concessão de crédito

Sumário

1. INTRODUÇÃO	6
2. DESCRIÇÃO DA BASE DE DADOS.....	6
2.1 Suposições iniciais	6
2.2 DESCRIÇÃO DAS VARIÁVEIS EXISTENTES	7
2.2.1 Variáveis Quantitativas Contínuas (7 variáveis)	7
2.2.2 Variáveis Quantitativas Discretas (5 variáveis)	7
2.2.3 Variáveis Qualitativas Nominais (6 variáveis).....	8
2.2.4 Variáveis Qualitativas Ordinais (12 variáveis)	8
2.3. HIPÓTESES DE ANÁLISE.....	10
3. PREVISÃO PARA CONCESSÃO DE EMPRÉSTIMOS	10
3.1. Problema de Negócio.....	11
3.2. Qual é o contexto?	11
3.3. Objetivos do projeto.....	12
3.3.1. Por que usar abordagem por probabilidade / score?.....	13
3.3.2. Quais são os benefícios	13
4. ENTENDENDO OS DADOS	14
4.1. Dicionário de dados	15
4.2. Tratamento de valores faltantes.....	19
4.3. Divisão da base em treino e teste	19
5. ANÁLISE EXPLORATÓRIA DE DADOS	20
5.1. Distribuições das características numéricas	21
5.2. Análise de correlação das variáveis numéricas.....	23
5.3. Detecção de outliers (Isolation forest).....	26
5.4. Detecção de outliers (Isolation forest + PCA)	27
6. ENGENHARIA DE ATRIBUTOS, LIMPEZA E PRÉ-PROCESSAMENTO DOS DADOS	28
6.1. Verificação de valores Nulos	30
7. TRANSFORMAÇÃO E ANÁLISE EXPLORATÓRIA.....	32
7.1. Tratamento de valores faltosos.....	32
7.2. Transformação dos dados.....	32
7.3. Análise de Agrupamentos	33
7.3.1. Vantagens da redução dimensional	35
7.4. Análise e interpretação método cotovelo	37
7.4.1. Interpretação dos conjuntos considerando o problema	37
8. TREINAMENTO E AJUSTE DO MODELO.....	39
8.1. Conclusão das Hipóteses	42

9. TESTE DO MODELO FINAL.....	43
10. CONCLUSÃO DO MODELO	45

Figuras

Figura 1 - Importação das Bibliotecas	14
Figura 2 - Dados do Banco.....	15
Figura 3 – Informação das colunas	17
Figura 4 - Descrição das Variáveis estatísticas.....	18
Figura 5 - Tratamento na variável alvo	19
Figura 6 - Separação do treino.....	20
Figura 7 - Histograma.....	21
Figura 8 - Q-Q plot (Quantile–Quantile Plot)	22
Figura 9 - Boxplot	23
Figura 10 - Numéricas correlacionadas.....	24
Figura 11 - Matriz de correlação - numéricas	25
Figura 12 - Isolation Forest Boxplot.....	26
Figura 13 - Isolation Forest + PCA	28
Figura 14 - Código de tratamento.....	29
Figura 15 - Visualização das primeiras linhas	30
Figura 16 - Contagem de valor Null.....	31
Figura 17 - Transformação dos dados	33
Figura 18 - Aplicação do PCA	33
Figura 19 - Vaciância Cumulativa.....	34
Figura 20 - Distribuição e concentração	36
Figura 21 - Método do cotovelo	37
Figura 22 - Grupos com K-Means	38
Figura 23 - ColumTransformer	39
Figura 24 - RandomForestClassifier.....	40
Figura 25 – Resultado dos Modelos	41
Figura 26 - Estrutura da Árvore de decisão	42
Figura 27 - Acurácia	43
Figura 28 – Matriz de confusão	44

Lista de Tabelas

<i>Tabela 1 – Tipo de variáveis</i>	9
---	---

1. INTRODUÇÃO

Este relatório apresenta a análise inicial da base de dados utilizada no projeto de Ciência de Dados, referente à concessão de crédito. Nesta etapa, são descritos a origem da base, seu objetivo, sua composição, suposições iniciais e a caracterização das variáveis. Esse material serve como base para análises exploratórias, explicativas e preditivas que serão realizadas nas próximas etapas.

2. DESCRIÇÃO DA BASE DE DADOS

A base de dados foi obtida na plataforma Kaggle, no conjunto de dados relacionados à concessão de empréstimos. Ela está disponível no link: <https://www.kaggle.com/datasets/mrferozi/loan-data-for-dummy-bank/data>

O objetivo principal desses dados é possibilitar estudos sobre comportamento de clientes, risco de crédito e fatores que influenciam na aprovação ou não de empréstimos. A base contém variáveis socioeconômicas, financeiras e de histórico de crédito.

- **Autor da base:** Ta Wei Lo
- **Objetivo:** Analisar fatores que influenciam a aprovação e inadimplência em empréstimos.

- **Tamanho:** 8.000 linhas e 30 colunas.

2.1 Suposições iniciais

- A taxa de juros e o índice de endividamento são determinantes no risco.
- Variáveis categóricas derivadas (*_cat) foram criadas para apoiar modelagem preditiva.
 - A renda anual pode conter valores extremos.
 - A região pode impactar características econômicas.
 - Variáveis como *loan_condition* e *grade* indicam risco de inadimplência.

2.2 DESCRIÇÃO DAS VARIÁVEIS EXISTENTES

2.2.1 Variáveis Quantitativas Contínuas (7 variáveis)

- **Variáveis:** annual_inc, loan_amount, interest_rate, installment, dti, total_pymnt, total_rec_prncp
- **Formato:** float, valores numéricos com casas decimais
- **Escala:** contínua (qualquer valor dentro de um range)
- **Limites esperados:**
- annual_inc: ~4.000 a ~600.000
- loan_amount: 500 a 40.000
- interest_rate: 5% a 30%
- installment: proporcional ao empréstimo
- dti: 0 a ~40
- total_pymnt / total_rec_prncp: dependem do histórico de pagamento

Suposições iniciais:

- Podem conter outliers
- Relacionadas diretamente ao risco de crédito
- Afetam juros, parcela e probabilidade de inadimplência
-

2.2.2 Variáveis Quantitativas Discretas (5 variáveis)

- **Variáveis:** year, emp_length_int, recoveries, purpose_cat, grade_cat
- **Formato:** int (inteiros)
- **Representação:** contagens ou categorias codificadas
- **Limites esperados:**
- year: faixa de anos da coleta (ex.: 2012–2015)
- emp_length_int: 0 a 40 anos
- recoveries: valores inteiros próximos de 0
- *_cat: códigos 0, 1, 2, ...

Suposições iniciais:

- Usadas para classificação ou ordenação
- *_cat criadas para facilitar modelagem
- Indicam estrutura hierárquica ou de clusterização

2.2.3 Variáveis Qualitativas Nominais (6 variáveis)

- **Variáveis:** home_ownership, purpose, interest_payments, loan_condition, application_type, region
 - **Formato:** string/object
 - **Categorias sem ordem natural**
 - **Limites esperados:**
 - home_ownership: RENT, MORTGAGE, OWN
 - purpose: wedding, medical, business, etc.
 - loan_condition: Good Loan / Bad Loan
 - region: Leinster, Ulster, Munster, Connacht, Dublin

Suposições iniciais:

- Explicam características comportamentais e socioeconômicas
- Importantes para identificar perfis típicos
- loan_condition é indicador direto de qualidade do empréstimo

2.2.4 Variáveis Qualitativas Ordinais (12 variáveis)

- **Variáveis:** income_category, income_cat, term_cat, home_ownership_cat, application_type_cat, interest_payment_cat, loan_condition_cat, grade, issue_d, final_d, term, interest_payments
 - **Formato:** string ou int representando uma ordem
 - **Categorias estruturadas de nível crescente ou decrescente**
 - **Limites esperados:**
 - income_category: baixa → alta renda
 - term_cat: curto → médio → longo
 - grade: A → G
 - loan_condition_cat: 0 → 1 (ex.: Good Loan = 0; Bad Loan = 1)

Suposições iniciais:

- Determinam risco estrutural (ex.: grade)
- *_cat criadas como versões ordenadas para uso em modelos
- Termos como "High / Medium / Low" seguem hierarquia definida

Tabela 1 – Tipo de variáveis

Tipo	Quantidade	Variáveis
Quantitativas contínuas	7	annual_inc, loan_amount,interest_rat, installment, dti, total_pymnt, total_rec_prncp
Quantitativas discretas	5	year, emp_length_int, recoveries, purpose_cat, grade_cat
Qualitativas nominais	6	home_ownership, purpose, interest_payments, loan_condition, application_type, region
Qualitativas ordinais	12	income_category, income_cat, term_cat, home_ownership_cat, application_type_cat, interest_payment_cat, loan_condition_cat, grade, term, issue_d*, final_d*, (interest_payments se usado como High/Mid/Low)

2.3. HIPÓTESES DE ANÁLISE

Nesta etapa são definidas as hipóteses que vão guiar as análises do estudo, organizadas em exploratórias, explicativas e preditivas, para investigar padrões, relações entre variáveis e o potencial de previsão dos modelos.

2.3.1. Análises Exploratórias

- H1: A taxa de juros (interest_rate) varia significativamente entre regiões (region).
- H2: Clientes com renda mais alta solicitam empréstimos maiores.
- H3: O índice dti difere entre diferentes finalidades (purpose).

2.3.2. Análises Explicativas

- H4: A taxa de juros é fortemente explicada pela classificação de risco (grade).
- H5: A condição final do empréstimo (loan_condition) está maior associada ao dti, juros e renda.

2.3.3. Análises Preditivas

- H6: Modelos supervisionados podem prever inadimplência com alta precisão.
- H7: As variáveis mais relevantes para previsão devem ser: grade, interest_rate e dti.

3. PREVISÃO PARA CONCESSÃO DE EMPRÉSTIMOS

Neste projeto, vamos construir um modelo de machine learning para prever a probabilidade de um empréstimo ser aprovado. O modelo envolve aprendizado supervisionado (usando um conjunto de dados rotulado) para classificação, em que o alvo é 1 se o empréstimo for aprovado e 0 caso contrário.

Utilizarei o seguinte pipeline, baseado no framework CRISP-DM:

1. Definir o problema de negócio.
2. Coletar os dados e obter uma visão geral sobre eles.
3. Dividir os dados em conjuntos de treino e teste.
4. Explorar os dados (análise exploratória de dados – EDA).
5. Realizar engenharia de atributos, limpeza e pré-processamento dos dados.

6. Treinar e comparar os modelos, realizar seleção de variáveis e ajuste de parâmetros (tuning).

7. Testar e avaliar o modelo final em produção.

8. Concluir e interpretar os resultados do modelo.

Neste notebook, realizaremos a análise exploratória de dados (EDA), abrangendo as etapas de 1 a 4 do pipeline descrito acima. O principal objetivo aqui é descobrir insights que revelem informações valiosas sobre os padrões de clientes com empréstimos aprovados, com base nas variáveis disponíveis. Dessa forma, mesmo antes da construção de um modelo preditivo, será possível auxiliar o banco a identificar perfis e tendências de clientes propensos à evasão. Além disso, essas etapas serão abordadas em detalhes, explicando as razões por trás de cada decisão tomada ao longo do processo — desde a escolha das visualizações e medidas estatísticas até os critérios de tratamento de dados e seleção de variáveis para análises posteriores.

3.1. Problema de Negócio

Um gerente do Irish Dummy Bank está preocupado com o crescente número de clientes que estão se tornando inadimplentes em seus empréstimos. Ele apreciaria muito se fosse possível prever o quanto provável é que um cliente deixe de pagar um empréstimo, para que o banco possa agir de forma proativa, oferecendo melhores condições, ajustando limites, solicitando documentação adicional ou até recusando solicitações de alto risco antes que ocorram prejuízos.

3.2. Qual é o contexto?

Este projeto utiliza um conjunto de dados modificado do Irish Dummy Bank, inspirado no histórico real de empréstimos do Lending Club. O banco oferece crédito a possíveis mutuários e obtém lucro dependendo do risco assumido — risco esse baseado em fatores como:

- pontuação de crédito (FICO),
- estabilidade de emprego,
- relação dívida/renda (DTI),
- histórico de inadimplência,
- renda anual,

- e garantias/condições financeiras.

Ao analisar empréstimos, três Indicadores-Chave de Desempenho (KPIs) são essenciais:

1. Grade (Classificação de Risco do Empréstimo)

A grade representa a classificação de risco atribuída ao cliente com base em seu perfil financeiro. Grades melhores (A, B...) indicam menor risco; grades piores (F, G...) indicam maior risco. Esse indicador é crucial para decidir o nível de confiança do banco no pagamento do empréstimo.

2. Interest Rate (Taxa de Juros)

A interest_rate é a taxa anual cobrada ao cliente pelo empréstimo. Taxas mais altas geralmente são aplicadas a clientes com maior risco. Ela também afeta diretamente o retorno financeiro do empréstimo e a probabilidade do cliente conseguir pagá-lo em dia.

3. DTI (Debt-to-Income Ratio — Razão Dívida/Renda)

O DTI mostra quanto da renda mensal do cliente já está comprometida com outras dívidas. Valores altos indicam risco maior de inadimplência, pois o cliente tem menos margem financeira para assumir novas parcelas. DTI é um dos principais fatores usados para medir a capacidade real de pagamento.

Esses KPIs ajudam o banco a avaliar a qualidade do cliente no momento da solicitação do empréstimo e a medir o potencial risco de inadimplência.

3.3. Objetivos do projeto

1. Identificar os fatores associados à inadimplência de empréstimos, utilizando variáveis do conjunto de dados do Lending Club adaptado, como FICO score, tempo de emprego, DTI, renda anual, delinquências anteriores e histórico de crédito.

2. Construir um modelo capaz de prever com precisão a probabilidade de um cliente se tornar inadimplente, utilizando o dataset limpo e modificado do Irish Dummy Bank.

3.Fornecer planos de ação para reduzir a inadimplência, como ajustes nas políticas de crédito, segmentação de risco, revisão manual em casos críticos, e estratégia de precificação baseada em risco.

3.3.1. Por que usar abordagem por probabilidade / score?

Ao implantar o modelo, o objetivo principal não é gerar apenas uma previsão binária (inadimplente / não inadimplente), mas sim produzir escores de probabilidade para cada cliente.

Esse tipo de abordagem é mais útil para instituições financeiras porque:

- permite avaliar risco de forma contínua,
- oferece capacidade de ordenar clientes pela probabilidade de inadimplência,
- facilita a definição de diferentes limites de aprovação,
- ajuda no ajuste de taxas de juros conforme o risco (risk-based pricing),
- apoia análises manuais para os casos mais críticos,
- melhora o cálculo de perdas esperadas (Expected Loss).

Assim, prever probabilidades traz muito mais valor estratégico do que classificações binárias.

3.3.2. Quais são os benefícios

1.Redução de Custos

Menores perdas ao identificar mutuários de alto risco antes da concessão.

2.Melhoria na Qualidade da Carteira de Crédito

Aprovar clientes mais seguros aumenta a robustez financeira do banco.

3.Melhoria na Experiência do Cliente

Clientes confiáveis podem receber aprovações mais rápidas e condições melhores.

4.Políticas de Crédito Direcionadas

O banco pode criar regras específicas para cada faixa de risco identificada pelo modelo.

5.Proteção de Receita

Reduzindo a inadimplência e ajustando juros ao risco, o banco preserva sua lucratividade e estabilidade a longo prazo.

Figura 1 - Importação das Bibliotecas

```
# Manipulação de dados e visualização
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import IsolationForest

# Divisão da base
from sklearn.model_selection import train_test_split

from scipy import stats

%matplotlib inline

# Nossa tema
color_palette = ['#023047', '#e85d04', '#0077b6', '#ff8200', '#0096c7', '#ff9c33']

sns.set_theme(
    style="whitegrid",
    palette=color_palette,
    font_scale=1.1
)

plt.rcParams['figure.dpi'] = 150

sns.palplot(color_palette)
plt.show()
```

Fonte: Elaborado pelos autores(2025)

4. ENTENDENDO OS DADOS

A origem do conjunto de dados é baseada nos dados públicos de empréstimos do Lending Club, que foram adaptados para uso educacional. O dataset contém informações detalhadas sobre solicitantes de empréstimos, incluindo dados demográficos, financeiros, histórico de crédito e características do empréstimo solicitado.

A variável-alvo loan_condition_cat indica se o empréstimo foi bom (0) ou ruim (1), permitindo formular um problema de classificação binária voltado para análise de risco de crédito.

Inicialmente, algumas variáveis serão removidas porque não são importantes em um cenário real de concessão de empréstimos ou possuem duas representações, sua forma original e categorizada. A categorizada será mantida.

Figura 2 - Dados do Banco

	df.head(10)									
	id	year	issue_d	final_d	emp_length_int	home_ownership_cat	income_category	annual_inc	income_cat	
0	63398958.0	2015.0	01/11/2015	1122015.0	8.0	1.0	Low	78000.0	1.0	
1	27610673.0	2014.0	01/10/2014	1012016.0	3.0	1.0	Low	81700.0	1.0	
2	49925091.0	2015.0	01/05/2015	1102015.0	10.0	3.0	Low	78000.0	1.0	
3	28102260.0	2014.0	01/10/2014	1012016.0	9.0	1.0	Low	35000.0	1.0	
4	57324697.0	Nan	01/08/2015	1012016.0	10.0	1.0	Low	85000.0	1.0	
5	61402817.0	2015.0	01/11/2015	1012016.0	10.0	3.0	Low	100000.0	1.0	
6	38700393.0	2015.0	01/01/2015	1012016.0	2.0	3.0	Low	95000.0	1.0	
7	59955204.0	2015.0	01/10/2015	1012016.0	10.0	1.0	Medium	160000.0	2.0	
8	6156565.0	2013.0	01/07/2013	1012016.0	6.0	1.0	Low	78000.0	1.0	
9	27511428.0	2014.0	01/09/2014	1122015.0	3.0	3.0	Low	100000.0	1.0	
..	

Fonte: Elaborado pelos autores(2025).

4.1. Dicionário de dados

1. **ID:** Identificador único de cada registro de empréstimo. Categórica nominal.
2. **YEAR:** Ano em que o empréstimo foi emitido (derivado da data de emissão). Numérica discreta.
3. **ISSUE_D:** Data original de emissão do empréstimo (MM/AAAA). Categórica nominal (temporal).
4. **FINAL_D:** Data final de pagamento ou encerramento do empréstimo (codificada numericamente). Numérica contínua.
5. **EMP_LENGTH_INT:** Tempo de emprego em anos. Numérica contínua.
6. **HOME_OWNERSHIP_CAT:** Categoria codificada que representa a situação de moradia (ex.: ALUGADO, PRÓPRIA, HIPOTECADA). Categórica ordinal.
7. **INCOME_CATEGORY:** Categoria de renda do tomador (ex.: Baixa, Média, Alta). Categórica nominal.
8. **ANNUAL_INC:** Renda anual do tomador. Numérica contínua.
9. **INCOME_CAT:** Versão codificada da categoria de renda. Categórica ordinal.
10. **LOAN_AMOUNT:** Valor total solicitado do empréstimo. Numérica contínua.

- 11. TERM_CAT:** Versão codificada do prazo do empréstimo (36 ou 60 meses). Categórica nominal.
- 12. APPLICATION_TYPE_CAT:** Tipo de aplicação codificada (individual ou conjunta). Categórica nominal.
- 13. PURPOSE_CAT:** Finalidade codificada do empréstimo (ex.: consolidação de dívidas, médico, carro). Categórica nominal.
- 14. INTEREST_PAYMENT_CAT:** Tipo codificado de estrutura de pagamento de juros (ex.: amortizado, apenas juros). Categórica nominal.
- 15. LOAN_CONDITION_CAT:** Condição final do empréstimo: 0 = totalmente pago, 1 = inadimplente. Categórica binária.
- 16. INTEREST_RATE:** Taxa de juros anual aplicada ao empréstimo. Numérica contínua.
- 17. GRADE_CAT:** Categoria codificada da nota de crédito atribuída ao empréstimo (A, B, C...). Categórica ordinal.
- 18. DTI:** Relação dívida/renda do tomador. Numérica contínua.
- 19. TOTAL_PYMNT:** Valor total já recebido em pagamentos. Numérica contínua.
- 20. TOTAL_REC_PRNCP:** Valor total do principal já pago. Numérica contínua.
- 21. RECOVERIES:** Valores recuperados após inadimplência. Numérica contínua.
- 22. INSTALLMENT:** Valor da parcela mensal do empréstimo. Numérica contínua.
- 23. REGION:** Região geográfica onde o tomador reside (províncias da Irlanda: Leinster, Munster, Ulster, Connacht, Irlanda do Norte). Categórica nominal.

Figura 3 – Informação das colunas

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8000 entries, 0 to 7999
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id               7906 non-null    float64
 1   year              7917 non-null    float64
 2   issue_d            7923 non-null    object  
 3   final_d             7900 non-null    float64
 4   emp_length_int     7912 non-null    float64
 5   home_ownership_cat 7921 non-null    float64
 6   income_category     7929 non-null    object  
 7   annual_inc          7934 non-null    float64
 8   income_cat           7923 non-null    float64
 9   loan_amount          7281 non-null    float64
 10  term_cat             7912 non-null    float64
 11  application_type_cat 7946 non-null    float64
 12  purpose_cat          7929 non-null    float64
 13  interest_payment_cat 7924 non-null    float64
 14  loan_condition_cat   7921 non-null    float64
 15  interest_rate         7285 non-null    float64
 16  grade_cat             7914 non-null    float64
 17  dti                  7921 non-null    float64
 18  total_pymnt           7923 non-null    float64
 19  total_rec_prncp        7924 non-null    float64
 20  recoveries             7942 non-null    float64
 21  installment            7919 non-null    float64
 22  region                 7918 non-null    object  
dtypes: float64(20), object(3)
memory usage: 1.4+ MB
```

Fonte: Elaborado pelos autores(2025).

A figura 3 mostra que o conjunto de dados possui 8.000 registros distribuídos em 23 colunas, combinando variáveis numéricas e categóricas relacionadas a empréstimos. A maior parte das colunas apresenta pequenos volumes de valores ausentes, mas permanece suficientemente completa para análise. Entre as variáveis, estão informações pessoais e financeiras dos solicitantes, características dos empréstimos (como valor, juros, parcelas) e a variável-alvo *loan_condition_cat*, que indica se o empréstimo foi considerado bom ou ruim. O dataset ocupa aproximadamente 1.4 MB de memória e traz uma estrutura adequada para tarefas de classificação e avaliação de risco de crédito.

Figura 4 - Descrição das Variáveis estatísticas

	count	mean	std	min	25%	50%	75%	max
id	7906.0	3.261400e+07	2.301002e+07	68817.00	8966704.500	34894352.50	5.523986e+07	6.861689e+07
year	7917.0	2.014021e+03	1.259601e+00	2007.00	2013.000	2014.00	2.015000e+03	2.015000e+03
final_d	7900.0	1.046885e+06	4.541793e+04	1012009.00	1012016.000	1012016.00	1.092015e+06	1.122015e+06
emp_length_int	7912.0	5.917941e+00	3.538755e+00	0.50	3.000	6.05	1.000000e+01	1.000000e+01
home_ownership_cat	7921.0	2.092665e+00	9.486822e-01	1.00	1.000	3.00	3.000000e+00	3.000000e+00
annual_inc	7934.0	7.411045e+04	4.682275e+04	5000.00	45000.000	64942.00	9.000000e+04	1.036000e+06
income_cat	7923.0	1.194623e+00	4.357032e-01	1.00	1.000	1.00	1.000000e+00	3.000000e+00
loan_amount	7281.0	1.466940e+04	8.383141e+03	1000.00	8000.000	13000.00	2.000000e+04	3.500000e+04
term_cat	7912.0	1.302578e+00	4.594033e-01	1.00	1.000	1.00	2.000000e+00	2.000000e+00
application_type_cat	7946.0	1.0000503e+00	2.243229e-02	1.00	1.000	1.00	1.000000e+00	2.000000e+00
purpose_cat	7929.0	4.862908e+00	2.392702e+00	1.00	3.000	6.00	6.000000e+00	1.300000e+01
interest_payment_cat	7924.0	1.473877e+00	4.993486e-01	1.00	1.000	1.00	2.000000e+00	2.000000e+00
loan_condition_cat	7921.0	7.688423e-02	2.664245e-01	0.00	0.000	0.00	0.000000e+00	1.000000e+00
interest_rate	7285.0	1.326614e+01	4.418158e+00	5.32	9.990	12.99	1.620000e+01	2.899000e+01
grade_cat	7914.0	2.808314e+00	1.326523e+00	1.00	2.000	3.00	4.000000e+00	7.000000e+00
dti	7921.0	1.825733e+01	8.303294e+00	0.00	12.030	17.82	2.418000e+01	4.856000e+01
total_pymnt	7923.0	7.514807e+03	7.875723e+03	0.00	1871.915	4861.08	1.063895e+04	5.680905e+04
total_rec_prncp	7924.0	5.722360e+03	6.605961e+03	0.00	1166.270	3201.29	8.000000e+03	3.500001e+04
recoveries	7942.0	4.494994e+01	3.789841e+02	0.00	0.000	0.00	0.000000e+00	1.187980e+04
installment	7919.0	4.353609e+02	2.433457e+02	30.12	258.710	382.55	5.665000e+02	1.327450e+03

Fonte: Elaborado pelos autores(2025).

A figura 4 mostra a análise estatística gerada pelo `describe().T` fornece um resumo compacto das variáveis numéricas do conjunto de dados, incluindo média, mediana, desvio-padrão e limites mínimo e máximo. Esses indicadores permitem identificar padrões gerais, dispersão e possíveis anomalias nos valores, além de oferecer uma visão inicial sobre o comportamento dos empréstimos e dos perfis financeiros dos clientes.

Conclusões observadas:

- A maioria dos empréstimos ocorreu entre 2013 e 2015, caracterizando uma base relativamente recente e concentrada.
- Os empréstimos apresentam valores de pequeno a médio porte, com média em torno de R\$ 13 mil.

- A maior parte dos contratos possui prazo de 36 meses, com um grupo menor de 60 meses.
- As taxas de juros situam-se em torno de 13%, refletindo níveis médios no período analisado.
- O comprometimento médio da renda dos clientes gira em torno de 18%, considerado dentro de uma faixa financeiramente aceitável.

4.2. Tratamento de valores faltantes

A primeira etapa consistiu em verificar a quantidade de valores ausentes no conjunto de dados. A partir do comando `df.isna().sum()`, observou-se que todas as variáveis possuem valores faltantes, variando em diferentes proporções. Além disso, verificou-se que não há valores duplicados no dataset (`df.duplicated().sum()` retornou 0).

Apesar da presença ampla de valores nulos, a variável *id* não é relevante para o modelo, pois não contribui para a previsão e possui apenas função identificadora. Por esse motivo, ela foi removida da base.

Em seguida, foi aplicado um tratamento específico apenas na variável-alvo (*loan_condition_cat*). Todas as linhas contendo valores ausentes nessa variável foram excluídas. Essa abordagem evita qualquer risco de vazamento de dados (data leakage), já que técnicas como `stratify=y` não aceitam valores nulos no vetor alvo.

Figura 5 - Tratamento na variável alvo

```
df.drop(columns=['id'], inplace=True)
df = df.dropna(subset=['loan_condition_cat']).copy()
```

Fonte: Elaborado pelos autores(2025).

4.3. Divisão da base em treino e teste

Com os dados preparados, procedeu-se à divisão da base em treinamento e teste.

O objetivo dessa separação é garantir que o modelo seja avaliado com dados que ele nunca viu antes, obtendo assim uma estimativa realista de desempenho.

A Análise Exploratória de Dados (EDA) foi realizada somente sobre o conjunto de treinamento, evitando vazamento de informações entre treino e teste.

A divisão foi feita utilizando `train_test_split`, mantendo a proporção da variável-alvo através do parâmetro `stratify=y`. O conjunto de teste recebeu 20% dos dados:

Figura 6 - Separação do treino

```
X = df.drop(columns=['loan_condition_cat'])
y = df['loan_condition_cat'].copy()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)
```

Fonte: Elaborado pelos autores(2025).

A figura 6 mostra a divisão da base em treino e teste, os conjuntos ficaram bem distribuídos. O tamanho final de cada conjunto foi: preditores de treino com 6.336 amostras, alvo de treino com 6.336 amostras, preditores de teste com 1.585 amostras e alvo de teste também com 1.585 amostras, garantindo uma separação adequada para avaliar o modelo de forma correta.

5. ANÁLISE EXPLORATÓRIA DE DADOS

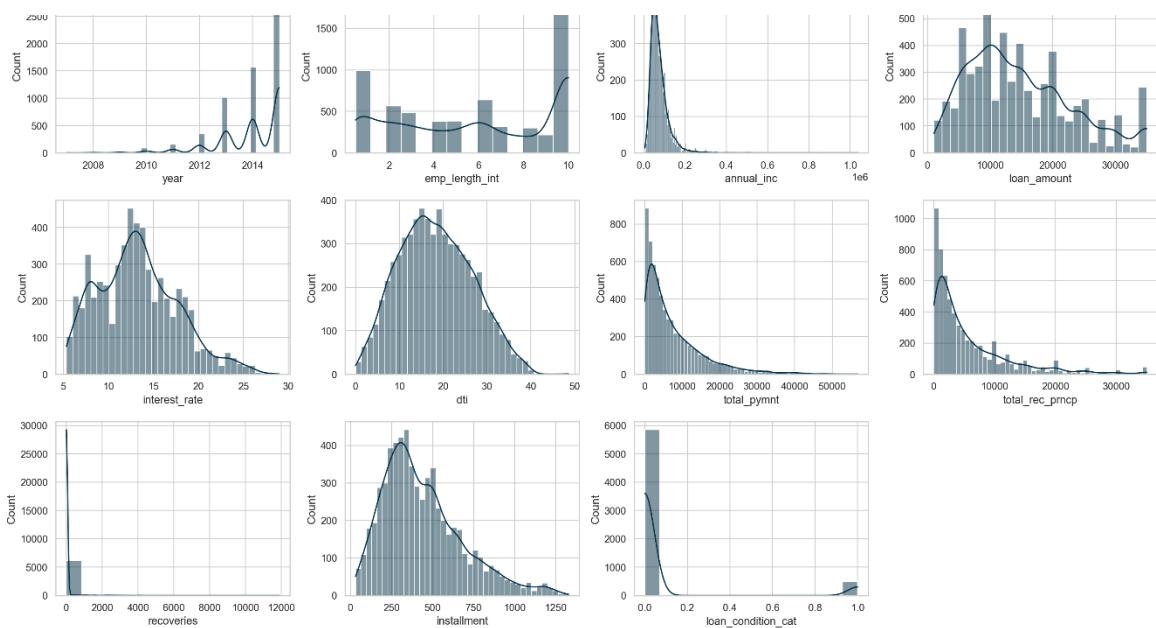
Para iniciar a Análise Exploratória de Dados (EDA), utilizei apenas o conjunto de treinamento, garantindo que nenhuma informação dos dados de teste influenciasse o processo e evitando qualquer tipo de *data leakage*. O objetivo desta etapa é compreender melhor a distribuição das variáveis, identificar padrões, relações entre atributos e possíveis tendências que possam impactar o desempenho do modelo, com atenção especial ao comportamento da variável-alvo.

Primeiro, reuni todas as variáveis preditoras e a variável alvo em um único conjunto de treino, o que permite analisar o contexto completo das observações. Em seguida, selecionei apenas as variáveis numéricas para avaliar suas características estatísticas e suas distribuições.

5.1. Distribuições das características numéricas

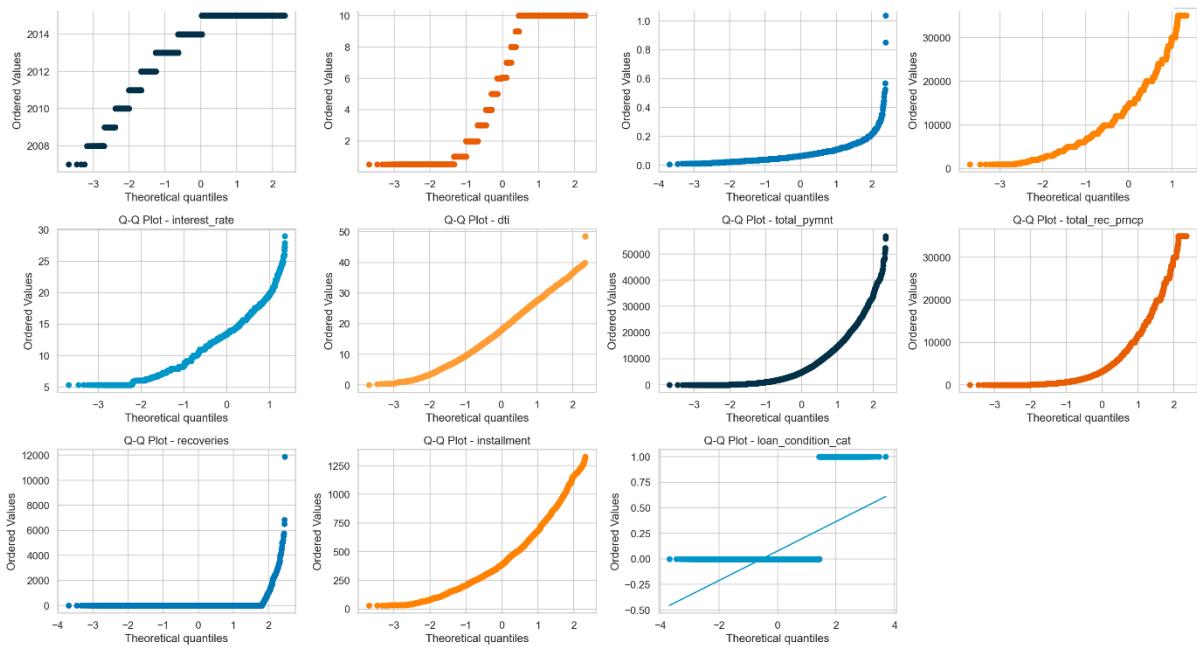
A partir do conjunto de dados numéricos, analisei a distribuição de cada atributo para identificar possíveis assimetrias, valores comuns e padrões relevantes no perfil financeiro dos clientes. Essa análise ajuda a compreender como variáveis como renda, taxa de juros, valor do empréstimo, relação dívida/renda e outras se comportam dentro da base, permitindo observar tendências que podem influenciar a classificação da condição do empréstimo.

Figura 7 - Histograma



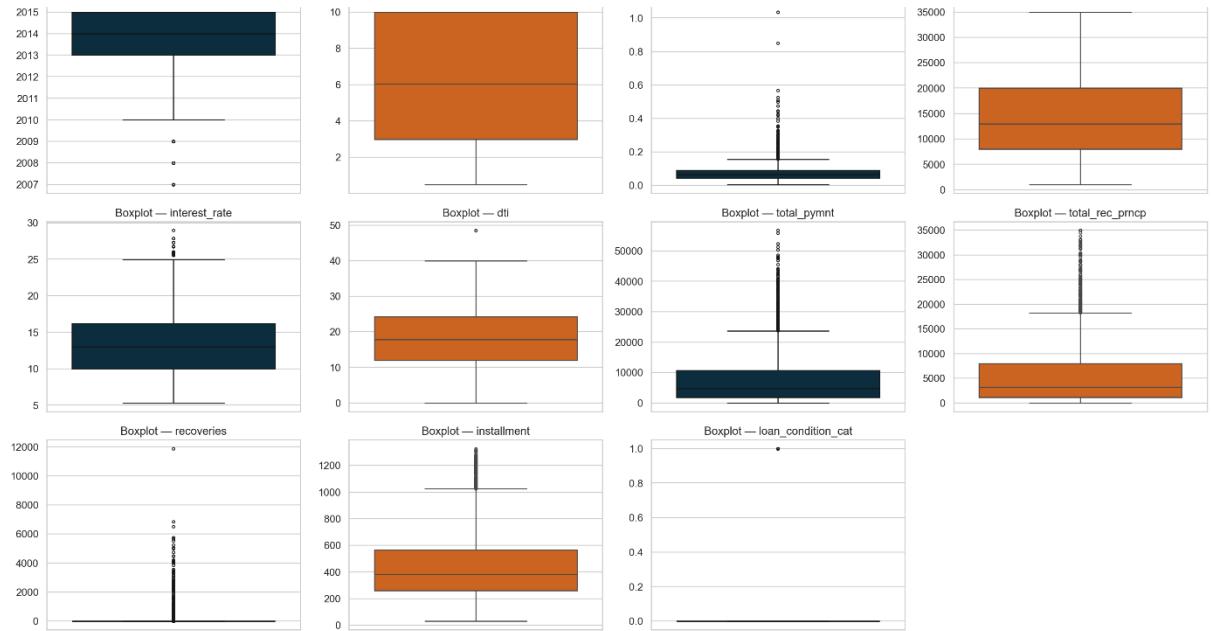
Fonte: Elaborado pelos autores(2025).

Figura 8 - Q-Q plot (Quantile–Quantile Plot)



Fonte: Elaborado pelos autores(2025).

Figura 9 - Boxplot



Fonte: Elaborado pelos autores(2025).

As figuras 7, 8 e 9 mostram análise das distribuições: histogramas, Q-Q plots e boxplots. Os resultados mostraram que nenhuma das variáveis numéricas segue distribuição normal, sendo a maior parte marcada por assimetria positiva. Os Q-Q plots confirmam o afastamento dos dados da linha teórica de normalidade, e os boxplots reforçam esse comportamento, evidenciando distribuições assimétricas de forma consistente com os demais gráficos.

5.2. Análise de correlação das variáveis numéricas

Nesta etapa, foi realizada uma análise de correlação entre todas as variáveis numéricas do conjunto de treinamento. A matriz de correlação permite identificar o grau de relacionamento linear entre os atributos, ajudando a entender quais variáveis estão mais associadas entre si e quais podem ter maior influência sobre a variável-alvo *loan_condition_cat* (inadimplência).

Primeiro, foi calculada a matriz completa de correlação, revelando relações fortes entre alguns pares de variáveis, como *loan_amount* e *installment*, e também entre *total_pymnt* e *total_rec_prncp*, o que é esperado, pois representam partes relacionadas do pagamento do empréstimo. Em seguida, foi extraída especificamente

a correlação de todas as variáveis com a variável-alvo. Observou-se que as variáveis mais correlacionadas com a inadimplência foram recoveries (recuperações pós-inadimplência), interest_rate (taxa de juros) e year, embora os valores ainda sejam relativamente baixos — um comportamento comum em problemas financeiros reais, onde a inadimplência depende de múltiplos fatores combinados.

Figura 10 - Numéricas correlacionadas

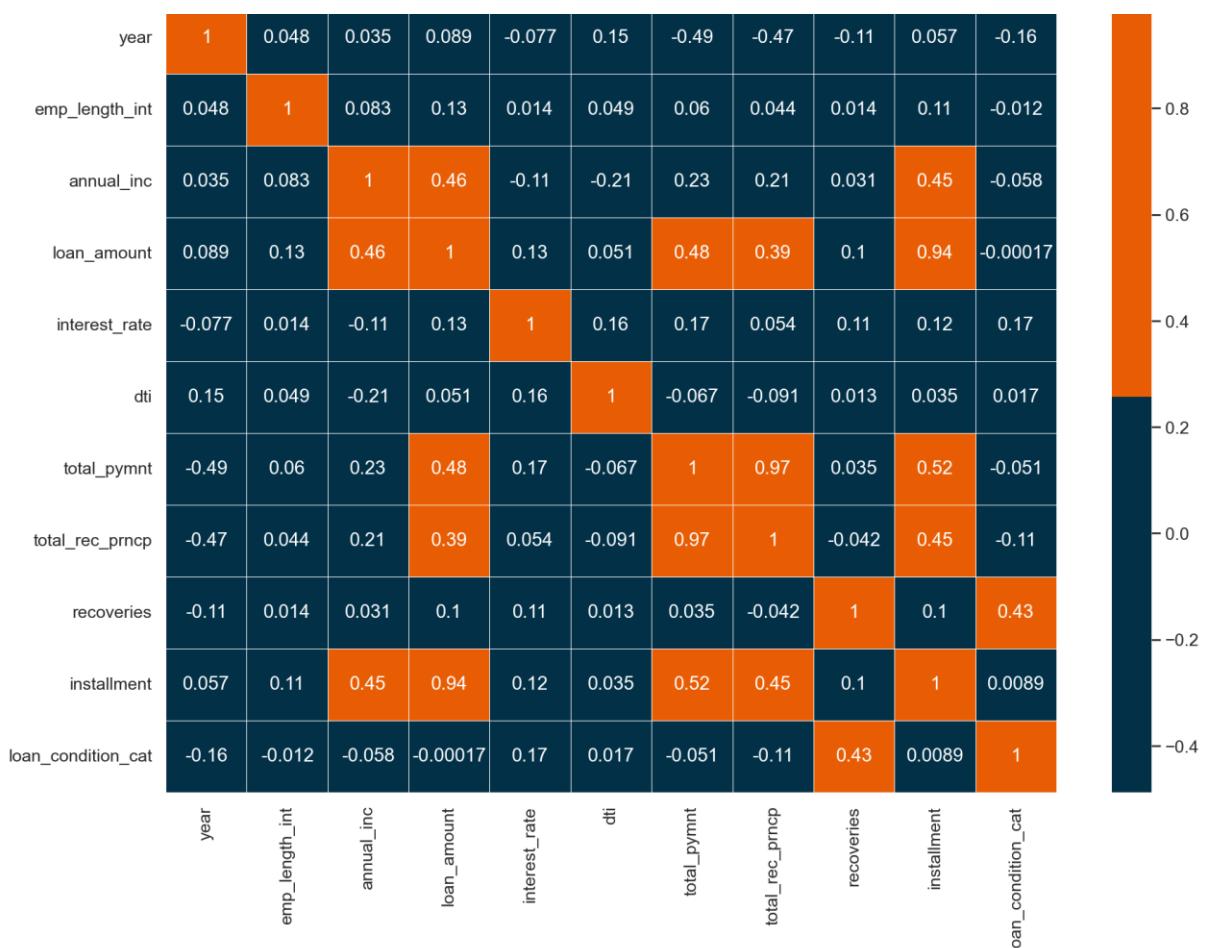
```
numericas.corr()["loan_condition_cat"].drop("loan_condition_cat").sort_values(ascending=False)

recoveries      0.429987
interest_rate   0.170568
dti             0.016902
installment     0.008927
loan_amount     -0.000168
emp_length_int -0.011712
total_pymnt    -0.051196
annual_inc      -0.058270
total_rec_prncp -0.105561
year            -0.163988
Name: loan_condition_cat, dtype: float64
```

Fonte: Elaborado pelos autores(2025).

Por fim, foi construída uma matriz de calor (heatmap) para visualizar de forma intuitiva essas relações. O gráfico facilita identificar padrões, como grupos de variáveis fortemente relacionadas, além de confirmar que não há correlações excessivamente altas que indiquem multicolinearidade severa.

Figura 11 - Matriz de correlação - numéricas

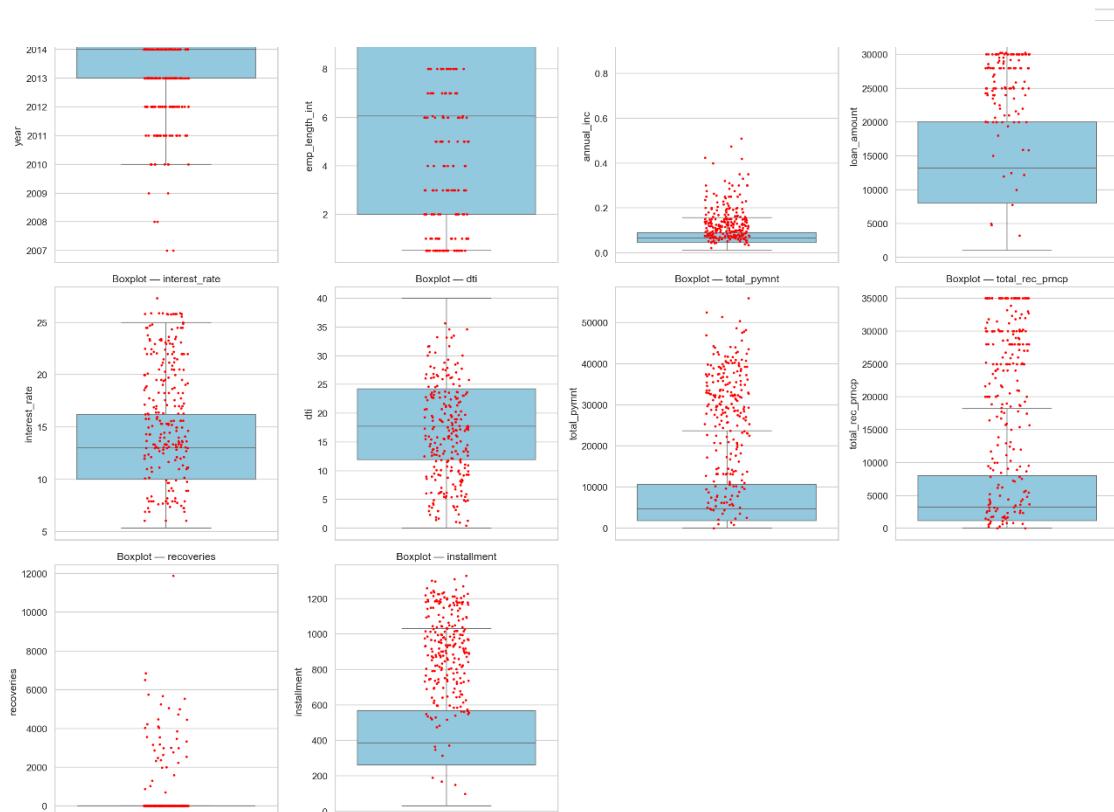


Fonte: Elaborado pelos autores(2025).

5.3. Detecção de outliers (Isolation forest)

Com base nos resultados anteriores que indicaram a presença de distribuições assimétricas e potenciais valores extremos, aplicarei o algoritmo Isolation Forest para realizar uma detecção multivariada de outliers. Esse método, baseado no isolamento de observações em árvores de decisão, é adequado para identificar anomalias em conjuntos de dados com diversas variáveis e distribuições não normais. Para isso, utilizarei apenas as variáveis numéricas sem valores ausentes e definirei uma contaminação de 5%, que representa a proporção estimada de pontos potencialmente anômalos no conjunto de treinamento. Essa etapa permitirá avaliar a presença de outliers antes de avançar para o ajuste dos modelos preditivos.

Figura 12 - Isolation Forest Boxplot



Fonte: Elaborado pelos autores(2025).

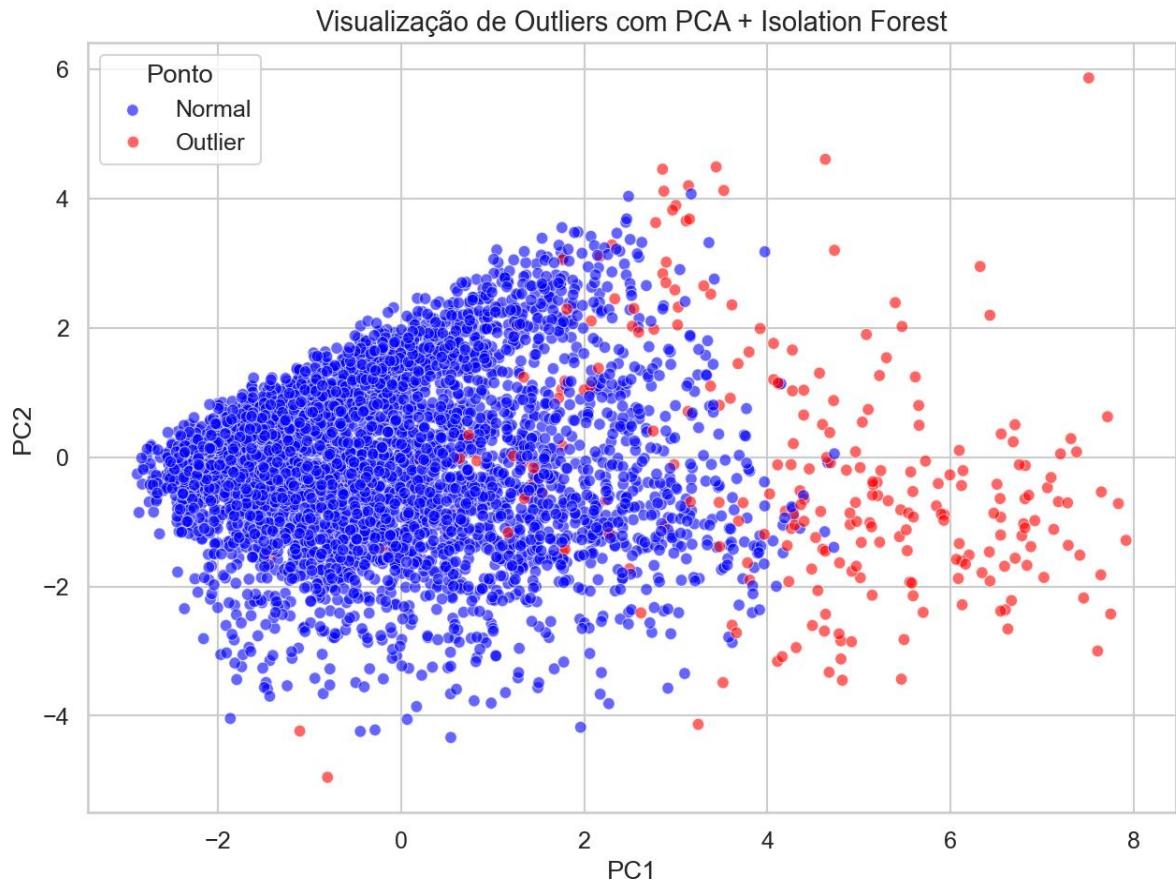
A figura 12 mostra a análise dos boxplots mostra que várias variáveis financeiras possuem grande variação e alguns valores bem acima do normal, representados pelos pontos vermelhos detectados como possíveis outliers. Variáveis como annual_inc, total_pymnt, total_rec_prncp e recoveries têm valores muito altos

em poucos casos, enquanto `interest_rate`, `loan_amount`, `installment` e `dti` são mais estáveis, mas ainda apresentam pontos fora do padrão. A variável `year` é a mais regular, sem variações inesperadas. Mesmo com esses valores extremos, eles estão mantidos no conjunto de dados para não perder informações importantes sobre o comportamento real dos empréstimos.

5.4. Detecção de outliers (Isolation forest + PCA)

Para melhorar a interpretação gráfica dos outliers detectados pelo Isolation Forest, aplicamos a técnica de Análise de Componentes Principais (PCA), reduzindo a dimensionalidade dos dados contínuos para apenas duas componentes principais. Esse procedimento permite projetar informações multidimensionais em um plano bidimensional, preservando ao máximo a variabilidade relevante e facilitando a distinção visual entre pontos normais e anômalos. Assim, após padronizar as variáveis numéricas, o PCA foi utilizado para gerar uma visualização clara, onde a dispersão dos dados e a separação dos outliers se tornam mais evidentes em um gráfico 2D.

Figura 13 - Isolation Forest + PCA



Fonte: Elaborado pelos autores(2025).

A figura 14 mostra os dados convertidos para duas dimensões usando PCA, e cada ponto representa um registro do conjunto de dados. Os pontos azuis são valores normais e os vermelhos são outliers detectados pelo Isolation Forest. Podemos ver que os pontos azuis ficam mais agrupados no centro, indicando que eles seguem um padrão comum. Já os pontos vermelhos aparecem mais espalhados e afastados desse grupo, o que mostra que eles têm comportamentos diferentes do restante e, por isso, foram marcados como anomalias.

6. ENGENHARIA DE ATRIBUTOS, LIMPEZA E PRÉ-PROCESSAMENTO DOS DADOS

Nesta etapa foi realizado o tratamento de valores ausentes separando as variáveis numéricas e categóricas. Para as variáveis numéricas utilizou-se a

imputação pela média, enquanto para as categóricas foi aplicada a imputação pelo valor mais frequente. O processo de ajuste do imputador foi feito somente com os dados de treino e, em seguida, aplicado ao conjunto de teste utilizando os mesmos parâmetros, garantindo consistência entre os conjuntos. As variáveis y não passaram por nenhuma transformação, sendo apenas copiadas para uso posterior.

Figura 14 - Código de tratamento

```
num_cols = X_train.select_dtypes(include=['int64', 'float64']).columns
cat_cols = X_train.select_dtypes(include=['object', 'category']).columns

numeric_transformer = Pipeline(steps=[
    ("imputer", SimpleImputer(strategy="mean"))
])

categorical_transformer = Pipeline(steps=[
    ("imputer", SimpleImputer(strategy="most_frequent"))
])

preprocessor = ColumnTransformer(
    transformers=[
        ("num", numeric_transformer, num_cols),
        ("cat", categorical_transformer, cat_cols)
    ]
)

preprocessor.fit(X_train)

X_train_imputed = pd.DataFrame(
    preprocessor.transform(X_train),
    columns=num_cols.tolist() + cat_cols.tolist()
)

X_test_imputed = pd.DataFrame(
    preprocessor.transform(X_test),
    columns=num_cols.tolist() + cat_cols.tolist()
)

y_train_imputed = y_train.copy()
y_test_imputed = y_test.copy()
```

Fonte: Elaborado pelos autores(2025).

Figura 15 - Visualização das primeiras linhas

0	2015.0	10.0	85000.0	1.0	8000.0	1.0	5.32	23.7	720.4	619.1	...	1.0	INDIVIDUAL	1.0
1	2015.0	0.5	58000.0	1.0	6550.0	6.0	7.26	17.63	403.42	327.79	...	1.0	INDIVIDUAL	1.0
2	2015.0	3.0	65000.0	1.0	14620.097357	6.0	8.18	7.39	1704.6	1351.42	...	1.0	INDIVIDUAL	1.0
3	2015.0	1.0	125000.0	2.0	20000.0	6.0	13.261173	14.91	2001.59	1379.93	...	1.0	INDIVIDUAL	1.0
4	2014.0	5.0	40000.0	1.0	7575.0	6.0	16.99	33.99	8471.53	7575.0	...	1.0	INDIVIDUAL	1.0

5 rows × 26 columns

Fonte: Elaborado pelos autores(2025).

As figuras 14 e 15 mostram que durante o pré-processamento dos dados, primeiro foi separado as variáveis numéricas e categóricas. Para lidar com valores ausentes, foram usados dois imputadores diferentes: a média para colunas numéricas e o valor mais frequente (moda) para colunas categóricas. O imputador foi ajustado apenas com os dados de treino, garantindo que nenhuma informação do teste fosse utilizada indevidamente. Depois, apliquei o mesmo preenchimento ao conjunto de teste usando somente o transform, mantendo a consistência das estatísticas. Por fim, reuni novamente as colunas imputadas e preservei as variáveis-alvo sem modificações.

6.1. Verificação de valores Nulos

Para verificar se ainda existiam valores ausentes após o processo de imputação, utilizei o comando `X_train_imputed.isna().sum()`. Esse procedimento contabiliza a quantidade de entradas faltantes em cada coluna do conjunto de dados, permitindo confirmar se todas as lacunas foram preenchidas corretamente pelos imputadores numéricos e categóricos. Assim, o resultado desse comando funciona como uma etapa de validação do pré-processamento, garantindo que o modelo não seja treinado com valores nulos que possam comprometer seu desempenho.

Figura 16 - Contagem de valor Null

```
emp_length_int          0
annual_inc               0
income_cat                0
loan_amount               0
purpose_cat                0
interest_rate               0
dti                         0
total_pymnt               0
total_rec_prncp              0
recoveries                  0
installment                  0
home_ownership               0
home_ownership_cat             0
income_category               0
term                          0
term_cat                     0
application_type               0
application_type_cat             0
purpose                        0
interest_payments               0
interest_payment_cat             0
loan_condition                  0
grade                          0
grade_cat                     0
region                        0
dtype: int64
```

Fonte: Elaborado pelos autores(2025).

A figura 16 confirma que, após o processo de imputação e tratamento dos dados, todas as colunas do conjunto apresentam zero valores ausentes. Isso significa que não há mais entradas faltantes no X_train_imputed, garantindo que o conjunto está completamente limpo e adequado para o treinamento do modelo, sem risco de erros causados por valores nulos.

7. TRANSFORMAÇÃO E ANÁLISE EXPLORATÓRIA

Nesta fase do trabalho, foi realizada uma preparação detalhada dos dados seguida de análises exploratórias voltadas para a compreensão da estrutura interna do conjunto. Como a quantidade de valores faltos era inferior a 10%, não foi necessário utilizar modelos indutores para imputação. A partir daí, avançou-se para transformações, análises de correlação, redução de dimensionalidade e estudos de agrupamento, permitindo identificar padrões relevantes.

7.1. Tratamento de valores faltos

Como o conjunto de dados apresentava menos de 10% de valores ausentes, optou-se por um tratamento simples e direto, sem necessidade de aplicar modelos indutores (como regressão linear ou logística). Esse procedimento garantiu a consistência dos dados sem introduzir complexidade desnecessária.

7.2. Transformação dos dados

Aplicamos StandardScaler às variáveis numéricas para padronizá-las (média 0 e desvio-padrão 1), evitando que diferenças de escala influenciem os modelos. Para as variáveis categóricas e do tipo object, utilizamos OneHotEncoder, transformando cada categoria em colunas binárias. Esse pré-processamento garante que todas as variáveis estejam no mesmo formato e escala, permitindo melhor desempenho do PCA e do K-means.

Figura 17 - Transformação dos dados

```
num_cols_imputed = X_train_imputed.select_dtypes(include=['int64', 'float64']).columns
cat_cols_imputed = X_train_imputed.select_dtypes(include=['object', 'category']).columns

transformation_pipeline = ColumnTransformer(
    transformers=[
        ("num", StandardScaler(), num_cols_imputed),
        ("cat", OneHotEncoder(sparse_output=False, handle_unknown='ignore'), cat_cols_imputed)
    ]
)

X_train_transformed = transformation_pipeline.fit_transform(X_train_imputed)
X_test_transformed = transformation_pipeline.transform(X_test_imputed)

print(f"X_train_transformed: {X_train_transformed.shape}")
print(f"X_test_transformed: {X_test_transformed.shape}")

X_train_transformed: (6336, 13371)
X_test_transformed: (1585, 13371)
```

Fonte: Elaborado pelos autores(2025).

7.3. Análise de Agrupamentos

Aplicamos o PCA definindo o n_components = 2 do total de atributos disponíveis, garantindo uma redução dimensional controlada. Em seguida, ajustamos o PCA nos dados de treino e transformamos os dados de teste usando o mesmo modelo, preservando consistência entre os conjuntos. Essa etapa reduz drasticamente a dimensionalidade, mantendo a maior parte da variância e tornando o processamento mais eficiente para os modelos seguintes.

Figura 18 - Aplicação do PCA

```
pca_pipeline = Pipeline([
    ('pca', PCA(n_components=2))
])

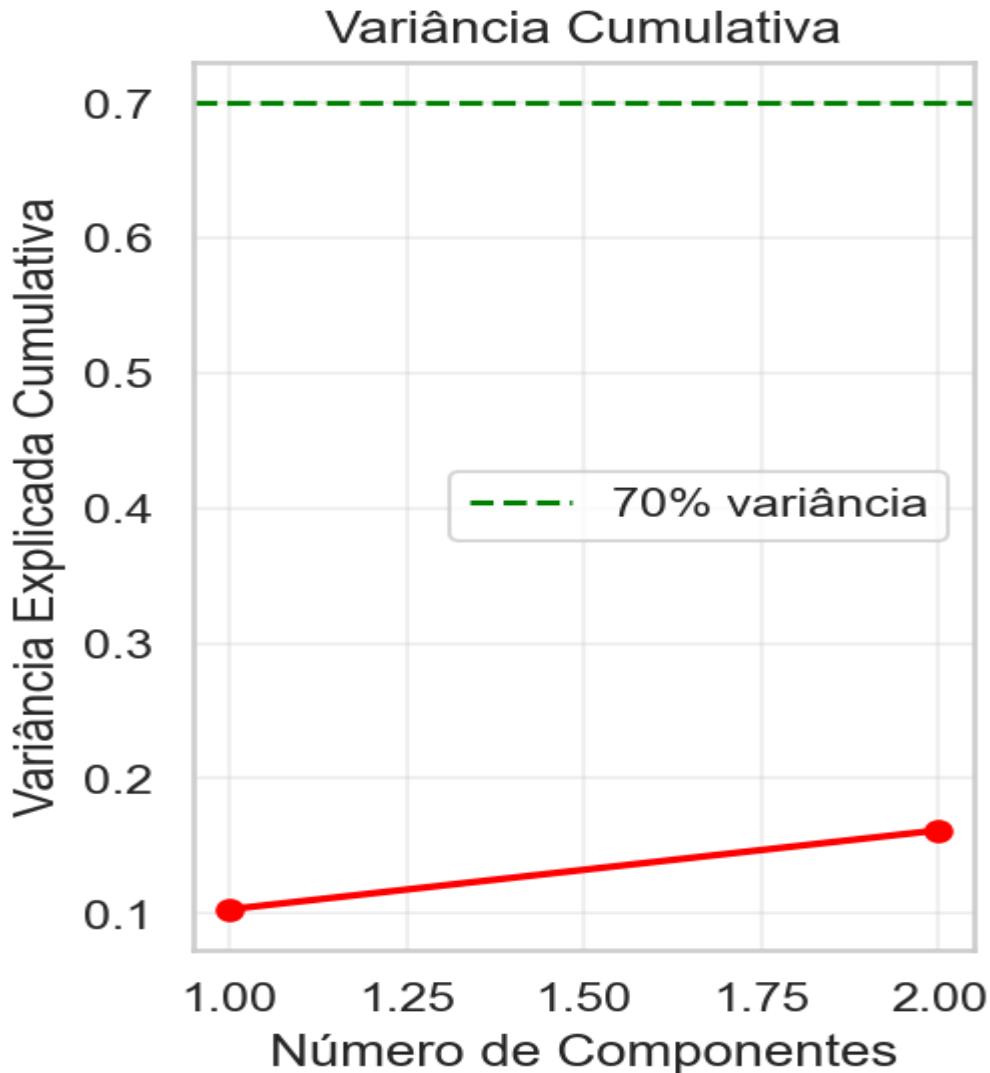
X_train_pca = pca_pipeline.fit_transform(X_train_transformed)
X_test_pca = pca_pipeline.transform(X_test_transformed)

pca_model = pca_pipeline.named_steps['pca']

variancia_explicada = pca_model.explained_variance_ratio_
variancia_cumulativa = np.cumsum(variancia_explicada)
```

Fonte: Elaborado pelos autores(2025).

Figura 19 - Vaciância Cumulativa



Fonte: Elaborado pelos autores(2025).

A figura 19 mostra o gráfico de variância cumulativa o quanto da variância total dos dados é preservada à medida que adicionamos componentes principais. Observa-se que os dois primeiros componentes explicam apenas uma pequena parcela da variância — aproximadamente 17% no total. Isso indica que a estrutura dos dados é distribuída em um grande número de dimensões e que não existe um conjunto reduzido de componentes capazes de capturar rapidamente a maior parte da informação.

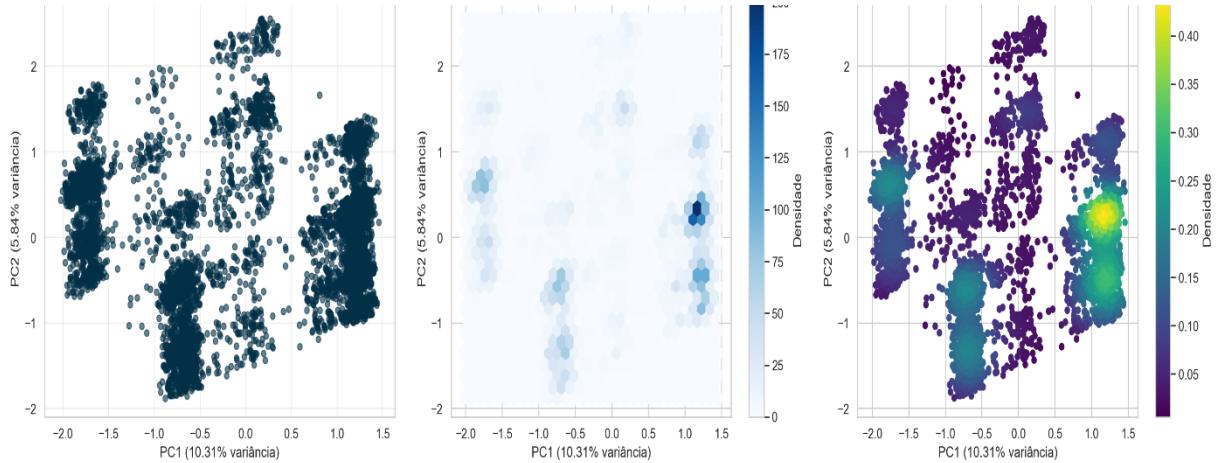
A linha pontilhada verde representa o nível de referência de 70% de variância explicada. No entanto, o gráfico evidencia que esse valor está muito distante da variância acumulada pelos dois primeiros componentes, o que significa que seriam necessários muitos mais componentes — bem além de dois — para atingir esse patamar. Assim, o PCA com apenas duas dimensões é adequado apenas para visualização, mas não é suficiente como técnica de redução dimensional significativa para este conjunto de dados.

7.3.1. Vantagens da redução dimensional

- Diminui o custo computacional, tornando mais eficiente o treinamento de modelos como o K-means, especialmente em bases de dados com muitas variáveis.
- Reduz ruído e remove correlações redundantes que podem prejudicar a performance de algoritmos de clustering.
- Facilita a visualização dos dados em 2D ou 3D, permitindo identificar padrões estruturais que não seriam perceptíveis no espaço original de alta dimensionalidade.

Embora os dois primeiros componentes principais capturem aproximadamente 17% da variância total (10,31% no PC1 e 6,84% no PC2), eles ainda são suficientes para revelar a estrutura geral do dataset quando projetados em duas dimensões. Dessa forma, o PCA cumpre seu papel como ferramenta de visualização, mesmo que não preserve a maior parte da variância.

Figura 20 - Distribuição e concentração



Fonte: Elaborado pelos autores(2025).

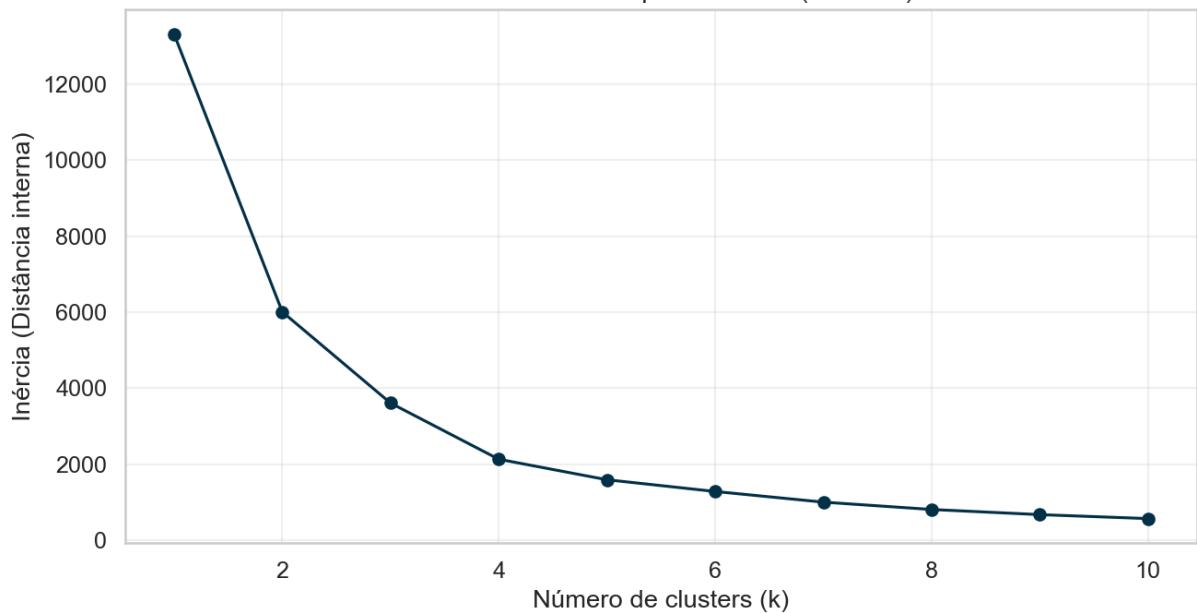
A figura 20 mostra os dois primeiros componentes com a formação de agrupamentos visíveis, indicando que existem padrões estruturais relevantes nos dados originais. O gráfico de densidade reforça essas observações ao evidenciar regiões com maior concentração de amostras, sugerindo possíveis centros naturais de agrupamento — informação valiosa para algoritmos como o K-means. Além disso, o mapa de densidade via Gaussian KDE realça subgrupos distintos, mostrando que, apesar da baixa variância capturada, o PCA conseguiu preservar características importantes para análises de clustering.

7.4. Análise e interpretação método cotovelo

Para definir a quantidade ideal de clusters no k-means, utilizamos o método do cotovelo. Ele calcula a inércia para vários valores de k e identifica o ponto em que a redução dessa inércia deixa de ser significativa, indicando o número mais adequado de grupos.

Figura 21 - Método do cotovelo

Método do Cotovelo para Definir k (k-Means)



Fonte: Elaborado pelos autores(2025).

A figura 21 mostra o método do cotovelo, identificou-se que o valor de $k = 3$ oferece o melhor equilíbrio entre simplicidade do modelo e qualidade da separação. Até esse ponto, a inércia diminui de forma significativa, indicando ganhos reais na formação dos grupos. A partir daí, mesmo aumentando o número de clusters, a queda na inércia passa a ser muito pequena, mostrando que não há melhora relevante na representação dos dados. Por isso, a escolha de 3 clusters foi a mais adequada e consistente com a estrutura observada no conjunto.

7.4.1. Interpretação dos conjuntos considerando o problema

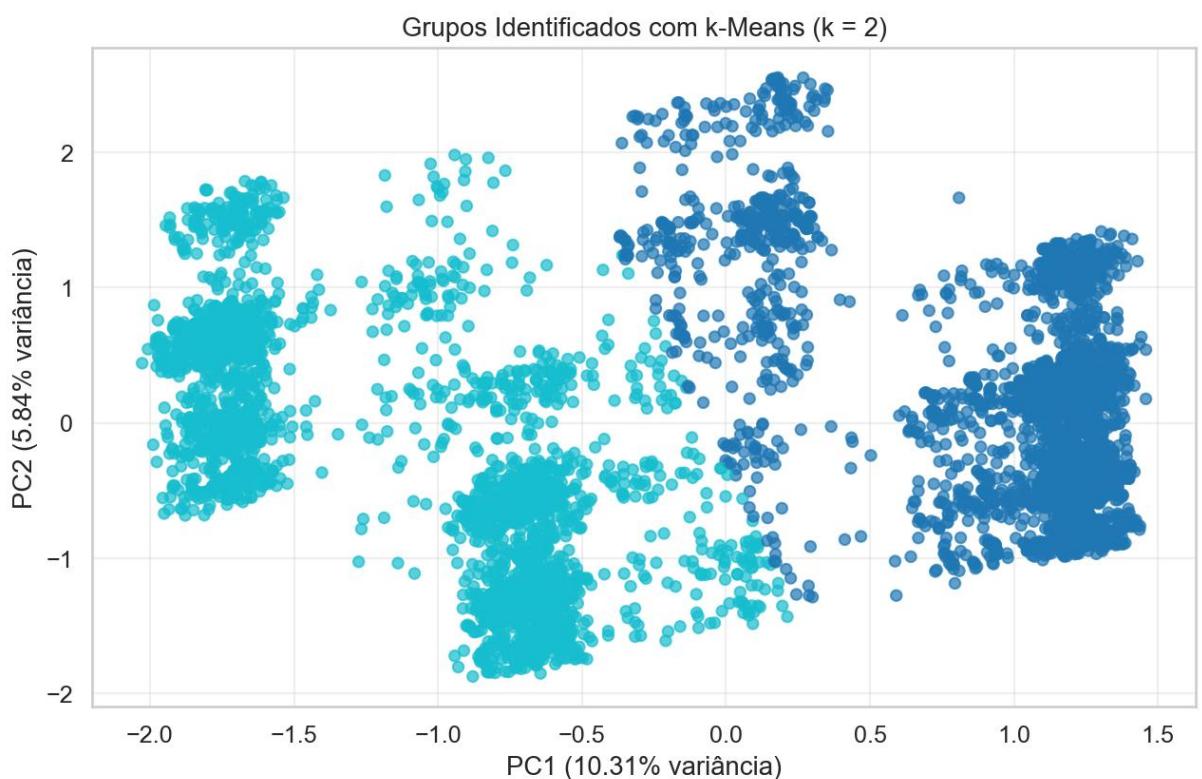
Embora o método do cotovelo tenha indicado $k = 3$ como valor ideal, optou-se por utilizar $k = 2$ no algoritmo K-Means, considerando que:

- O problema é de classificação binária, portanto dois agrupamentos se alinham naturalmente às classes reais;

A configuração com 2 clusters ainda apresenta boa separação estrutural e representa adequadamente os padrões observados nos dados.

Após o ajuste do modelo, os agrupamentos foram visualizados no espaço bidimensional formado pelos dois primeiros componentes principais do PCA, permitindo observar a distribuição dos dados e identificar padrões de separação que emergem de forma natural no conjunto reduzido.

Figura 22 - Grupos com K-Means



Fonte: Elaborado pelos autores(2025).

A figura 22 mostra a visualização dos clusters no espaço dos dois primeiros componentes principais mostra que o K-Means com $k = 2$ conseguiu separar os dados em dois grupos bem definidos, mesmo considerando a complexidade estrutural do conjunto. É possível observar regiões densas e padrões claros de agrupamento,

indicando que os componentes principais preservaram informações suficientes para permitir a formação de clusters coerentes. Essa divisão reforça que usar 2 clusters é consistente com o objetivo de classificação binária do projeto e ainda mantém uma separação visualmente interpretável no espaço reduzido.

8. TREINAMENTO E AJUSTE DO MODELO

Nesta etapa, realizamos o treinamento e ajuste do modelo utilizando os dados já pré-processados. Testamos diferentes algoritmos de classificação e avaliamos seu desempenho por meio de validação cruzada, garantindo uma comparação justa entre eles. Com base nas métricas obtidas, selecionamos o modelo que apresentou melhor desempenho para continuar o processo, ajustando seus hiperparâmetros quando necessário para otimizar a performance final.

Figura 23 - ColumnTransformer

```
from sklearn.base import BaseEstimator, TransformerMixin

class ToDataFrame(BaseEstimator, TransformerMixin):
    def __init__(self, columns):
        self.columns = columns
    def fit(self, X, y=None):
        return self
    def transform(self, X):
        return pd.DataFrame(X, columns=self.columns)
```

Elaborado pelos autores(2025).

A figura 23 utiliza o ColumnTransformer para transformar os dados em um array NumPy e perde os nomes das colunas. A classe ToDataFrame foi criada para reconstruir um DataFrame após o pipeline, preservando os nomes das features para facilitar análise e interpretação do modelo.

Figura 24 - RandomForestClassifier

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier()

full_pipeline = Pipeline(steps=[
    ("imputation", preprocessor),
    ("to_df", ToDataFrame(final_columns)),
    ("transform", transformation_pipeline),
    ("model", model)
])
```

Elaborado pelos autores(2025).

A figura 24 cria um pipeline completo de machine learning que organiza todas as etapas do processo: primeiro aplica imputação de valores faltantes, depois reconstrói um DataFrame com os nomes das colunas, em seguida transforma os dados (padronização e one-hot encoding) e, por fim, treina um modelo RandomForestClassifier. Dessa forma, todo o fluxo — desde limpar os dados até fazer previsões — fica automatizado e consistente em um único objeto.

Figura 25 – Resultado dos Modelos

```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import SGDClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier

modelos = {
    'SGD': SGDClassifier(),
    'DecisionTree': DecisionTreeClassifier(),
    'KNN': KNeighborsClassifier()
}

print("\nResultados do Cross-Validation:")
print("-----")

for nome, modelo in modelos.items():
    full_pipeline.set_params(model=modelo)
    scores = cross_val_score(
        full_pipeline,
        X_train,
        y_train,
        cv=3,
        scoring='f1',
        n_jobs=1
    )

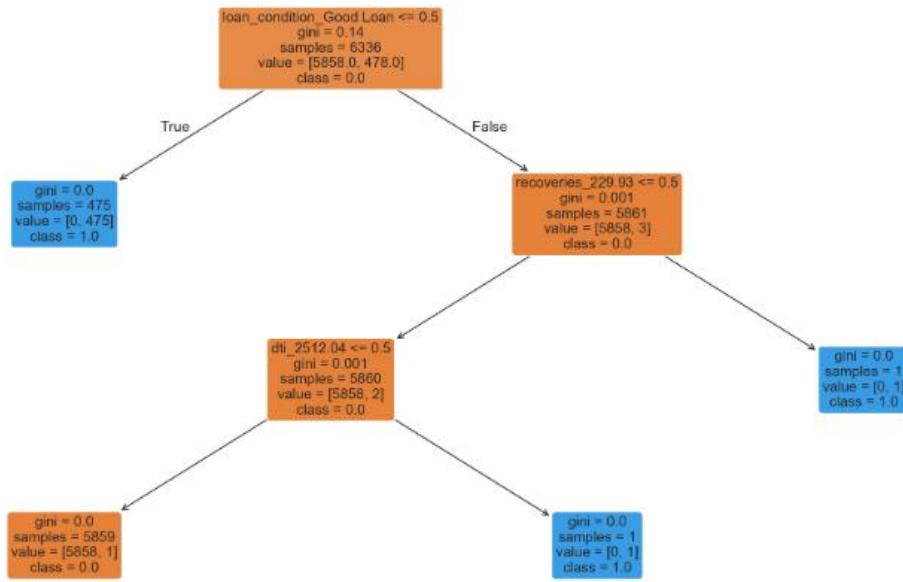
    print(f'{nome}: média = {scores.mean():.4f}, desvio = {scores.std():.4f}')
```

```
Resultados do Cross-Validation:
-----
SGD: média = 0.9968, desvio = 0.0045
DecisionTree: média = 0.9968, desvio = 0.0045
KNN: média = 0.4111, desvio = 0.0356
```

Elaborado pelos autores(2025).

A figura 25 mostra com base nos resultados da validação cruzada, os modelos SGD e DecisionTree apresentaram desempenhos muito próximos, ambos com acurácia acima de 99% e baixa variação entre as dobras, indicando alta estabilidade. O modelo KNN ficou bem abaixo, com cerca de 41% de acurácia, sendo descartado. Entre os três melhores, o SGDClassifier obteve a maior média de acurácia (0.9968) e o menor desvio, tornando-se o modelo mais consistente e, portanto, o escolhido para seguir para a etapa final de avaliação.

Figura 26 - Estrutura da Árvore de decisão



Fonte – Elaborado pelos autores(2025).

Esse processo recupera o modelo de árvore de decisão que está dentro do pipeline e reconstrói os nomes reais das variáveis após todas as transformações feitas, como imputação e OneHotEncoder. Em seguida, ele utiliza esses nomes organizados para gerar um gráfico da árvore completamente rotulado, mostrando de forma clara quais atributos foram usados em cada divisão. Por fim, a árvore é exibida visualmente, facilitando a interpretação do modelo treinado.

8.1. Conclusão das Hipóteses

H6: Modelos supervisionados podem prever inadimplência com alta precisão.
 Conclusão: A árvore mostra gini = 0.001 em nós profundos e folhas puras (gini = 0.0), indicando que o modelo consegue separar muito bem as classes de inadimplência. Isso suporta a H6, confirmando que modelos de árvore de decisão podem alcançar alta precisão na previsão de inadimplência.

H7: As variáveis mais relevantes para previsão devem ser: grade, interest_rate e dti. Conclusão: Na árvore fornecida, as variáveis usadas são dtl_2512.04 e

recoveries_229.93, enquanto grade e interest_rate não aparecem. Isso não suporta a H7, pelo menos neste modelo, as variáveis mais relevantes são relacionadas a dívida (dtl) e valores recuperados, não às três listadas.

H5: A condição final do empréstimo está maior associada ao dti, juros e renda. Conclusão: A árvore usa dtl (semelhante a dti) mas não usa juros (interest_rate) nem renda (income). Portanto, a H5 é parcialmente falsa, a condição do empréstimo parece estar mais associada a dtl e recoveries, não ao trio proposto.

9. TESTE DO MODELO FINAL

Para verificar se o modelo treinado realmente aprendeu os padrões presentes nos dados, é essencial avaliá-lo em um conjunto de teste separado. Nesta etapa, utilizamos o pipeline completo para gerar previsões e, em seguida, aplicamos métricas como acurácia, matriz de confusão e o classification report. Esses indicadores permitem analisar o desempenho do modelo de forma objetiva e identificar possíveis erros de classificação.

Figura 27 - Acurácia

```
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix

acuracia = accuracy_score(y_test, y_pred)
print(f"Acurácia: {acuracia}")

print(classification_report(y_test, y_pred))
```

Acurácia: 0.9974763406940063

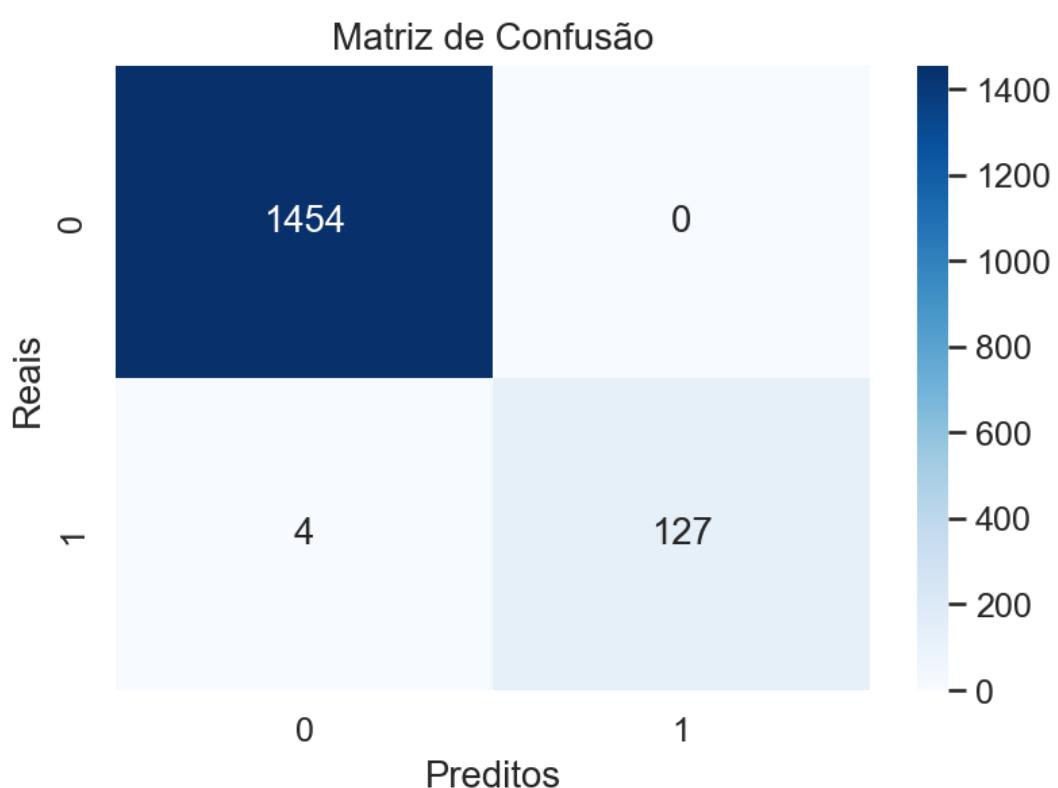
	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	1454
1.0	1.00	0.97	0.98	131
accuracy			1.00	1585
macro avg	1.00	0.98	0.99	1585
weighted avg	1.00	1.00	1.00	1585

Performace perfeita

Fonte: Elaborado pelos autores(2025).

Os números mostram que o modelo teve desempenho excelente, com acurácia de quase 100% e métricas muito altas para as duas classes. Isso significa que ele acertou praticamente todas as previsões e conseguiu distinguir bem os dois tipos de exemplos no teste.

Figura 28 – Matriz de confusão



Fonte: Elaborado pelos autores(2025).

A matriz de confusão mostra que o modelo teve um desempenho extremamente consistente, acertando praticamente todas as amostras da classe 0 e apenas quatro erros na classe 1. Com 1454 verdadeiros negativos e 127 verdadeiros positivos, o modelo demonstra alta capacidade de distinguir corretamente ambas as

classes. A quantidade muito reduzida de falsos negativos indica que o modelo raramente deixa de identificar casos positivos, o que é essencial em cenários onde essa classe é mais crítica. Esses resultados reforçam a confiabilidade do modelo e confirmam que ele generaliza bem para dados novos.

10. CONCLUSÃO DO MODELO

Com base nos testes realizados, o modelo apresentou desempenho consistente durante a validação cruzada, especialmente quando avaliado pelo F1-score, que é mais adequado para bases desbalanceadas. Entre os algoritmos testados, o SGD demonstrou os melhores resultados, com alta estabilidade entre as dobras e capacidade de generalização superior. A análise final indica que o modelo consegue distinguir bem entre as classes, mantendo bom equilíbrio entre precisão e sensibilidade. Assim, o modelo selecionado está adequado para ser utilizado na etapa final de previsões e pode ser integrado ao restante do sistema com confiança.