Previsão de Inadimplência para Concessão de Crédito

💡 Objetivo

Desenvolver um **modelo preditivo** para **estimar a probabilidade de inadimplência de pagamentos**. Espera-se, assim, fornecer **subsídio para a tomada de decisão envolvendo concessão de crédito**, ajudando a identificar clientes com maior probabilidade de não pagar uma cobrança no prazo esperado e, assim, definir ações proativas de cobrança para reduzir o índice de inadimplência.

📁 Fontes de Dados Utilizadas

Foram empregadas quatro bases de dados fornecidas para a construção e validação do modelo. A relação entre as bases se dá principalmente pelas chaves **ID_CLIENTE** e **SAFRA_REF**.

1. **base_cadastral:** informações cadastrais dos clientes, como porte, segmento industrial, CEP, e-mail e data de cadastro. Cada linha representa um cliente único (ID_CLIENTE) e seus dados não mudam ao longo do tempo (princípio da atemporalidade).
2. **base_info:** dados atualizados mensalmente com informações como renda do mês anterior e número de funcionários. Cada linha representa um cliente em um determinado mês (ID_CLIENTE, SAFRA_REF).
3. **base_pagamentos_desenvolvimento:** histórico de transações dos clientes, contendo a data de vencimento, valor a pagar, taxa e, quando disponível, a data de pagamento.
    - fundamental para o desenvolvimento do modelo, pois contém os dados de pagamento necessários para construir a variável *target.*
4. **base_pagamentos_teste:** contém as transações mais recentes, para as quais o modelo deve prever a probabilidade de inadimplência.
    - estrutura idêntica a da base de desenvolvimento, exceto pela DATA_PAGAMENTO.

🔍 Análise Exploratória e Limpeza dos Dados

Todas as bases estão anonimizadas, em conformidade com a LGPD. O processo completo de tratamento das bases pode ser consultado no **Anexo I**: análise, tratamento de valores nulos, identificação e remoção de outliers, assim como as justificativas das decisões tomadas.

🛠️ Engenharia de Features

Visando **extrair o máximo valor dos dados** e criar variáveis mais informativas, que possam enriquecer o modelo, a **base_pagamentos_desenvolvimento** foi enriquecida com Features da **base_info** e **base_cadastral**, por meio de operações de *joins*, conectando assim os dados de pagamentos, informações cadastrais e dados financeiros, utilizando as chaves para garantir a integridade e consistência dos registros.

- **base_info:** junção através de `PK_CLIENTE_SAFRA` (combinação de ID_CLIENTE e SAFRA_REF).
- **base_cadastral:** junção através de `ID_CLIENTE`.

Foram desenvolvidas as seguintes *features*:

- **ATRASO**: nº de dias de atraso no pagamento e base para cálculo do *target*.
- `INADIMPLENTE`: variável *target* binária, definida como **1 (inadimplente) se o atraso no pagamento for igual ou superior a 5 dias** e **0**, caso contrário.
- **PRAZO_CREDITO_DIAS**: Diferença em dias entre a data de emissão e a data de vencimento de um pagamento.
- **TEMPO_CADASTRO_ANOS**: tempo, em anos, desde o cadastro do cliente na base de dados
- **SAFRA_ANO**: ano extraído de **SAFRA_REF**.
- **SAFRA_MES**: mês extraído de **SAFRA_REF**.

Houveram dúvidas quanto à **granularidade** da modelagem: se deveria a **nível cliente** ou a **nível operação**. Optou-se, então, por **agrupar os dados com base na combinação de** `ID_CLIENTE` **e** `SAFRA_REF`.

⚙️ Modelagem Preditiva

Verificou-se que a necessidade de negócio e estrutura do *target* indicam um **problema de classificação binária** e aprendizado **supervisionado**.

Os modelos de **Random Forest Classifier**, **Regressão Logística**, **Árvore de Decisão (Decision Tree)**, **SVM**, **XGBoost**, **LightGBM** e **CatBoost**, dentre outros, podem ser usados para lidar com esse tipo de problema, por atuarem em classificação com dados tabulares e contam com mecanismos para reduzir o **overfitting**, que foi um ponto de atenção.

Neste projeto, foi selecionado o **Random Forest Classifier** por ser um modelo robusto, de fácil implementação e interpretação, com boa performance mesmo sem muitos ajustes finos. Por combinar várias árvores de decisão, oferece um equilíbrio eficaz entre **bias** e **variância**, sendo ideal para problemas de classificação com dados tabulares estruturados.

🧹 Pré-processamento de Dados

Para assegurar reprodutibilidade e consistência nas transformações, a etapa de pré-processamento foi organizada de forma que parte dela ocorre antes e parte dentro do **Pipeline** (classe do Scikit-learn).

- **OneHotEncoder para CEP_2_DIG:** essa variável categórica passou por uma etapa anterior de redução devido ao grande número de categorias, mantendo somente as mais comuns e substituindo as demais por OUTROS, e então foram codificadas.
- **StandardScaler para algumas variáveis numéricas**: padronização para garantir que nenhuma característica dominasse o aprendizado devido à sua escala. Algumas variáveis numéricas não sofrerão essa transformação por já estarem padronizadas.
- `PORTE` e `FLAG_PF:` foram submetidas a um processo de codificação manual anterior para um controle mais refinado da representação.
    - **Exemplo:** a codificação em `PORTE` deve ser diretamente proporcional ao tamanho da empresa, para os clientes pessoa jurídica.
- As colunas a seguir foram removidas por não serem consideradas relevantes para o treinamento do modelo, após uma análise crítica do negócio:
    - `PK_CLIENTE_SAFRA` e `ID_CLIENTE:` dados relacionados a identidade do cliente.
    - `SAFRA_REF:` substituído por `SAFRA_MES` e `SAFRA_ANO.`
    - `ATRASO:` usada apenas para calcular o *target.*
    - `DOMINIO_EMAIL:` o domínio de e-mail do cliente não afeta a inadimplência.
    - `DDD:` por estar relacionado à região geográfica, o CEP já cumpre essa função. Além disso, a sua inclusão poderia causar multicolinearidade.

Treinamento e Validação do Modelo: para assegurar a generalização do modelo e prevenir o *overfitting*, o treinamento e a validação do modelo incluíram a avaliação do desempenho sob diversas perspectivas.

Divisão Treino/Teste: os dados foram divididos em 80% para treinamento e 20% para teste.

📝 Métricas de Avaliação

A avaliação do desempenho do modelo foi realizada utilizando um conjunto abrangente de métricas:

- **Acurácia (Treino/Teste)**: mede a precisão do modelo tanto na fase de treinamento quanto na de teste.

- **Validação Cruzada com 5 Folds**: avaliação mais robusta do desempenho e para reduzir a variância na estimativa de erro .
    - Optou-se pela divisão do conjunto de dados em 5 partes (Folds), sendo que em cada rodada uma parte é usada para teste e as demais para treino — por oferecer um bom equilíbrio entre custo computacional e robustez na avaliação do modelo, permitindo estimar seu desempenho de forma mais confiável em diferentes subconjuntos dos dados.

- **Curva de Aprendizado**: análise do comportamento do modelo em relação ao tamanho do conjunto de treinamento, permitindo identificar indícios de *overfitting* ou *underfitting.*

- **Matriz de Confusão**: visualização dos acertos e erros do modelo.

- **Relatório de Classificação**: fornecimento de insights sobre a capacidade do modelo de identificar corretamente as classes positivas (inadimplência).

- **Curva ROC e AUC (área sob a curva)**:
    - A Curva ROC ilustra o equilíbrio entre Verdadeiros Positivos e Falsos Positivos.
    - A área sob a curva ROC é numericamente igual à capacidade do modelo de distinguir entre as classes, com valores próximos de 1 indicando melhor desempenho.

- **Estatística de Kolmogorov-Smirnov (KS):** métrica padrão ouro para modelos de classificação de crédito e amplamente usada em modelos de crédito para avaliar o poder de discriminação entre classes. Valores mais altos indicam que o modelo separa bem as duas classes — geralmente, valores acima de 0,6 são considerados bons no contexto financeiro.

📈 Resultados Alcançados

Os resultados obtidos com o modelo **Random Forest Classifier** foram promissores, demonstrando a capacidade preditiva da solução no contexto de risco de crédito:

Acurácia (Treino/Teste)

- **Acurácia no conjunto de treino**: 1.
- **Acurácia no conjunto de teste**: 0.9295.
- Apesar da acurácia perfeita no treino, a acurácia no conjunto de teste indica que o modelo mantém uma boa capacidade de generalização para dados não vistos.

⚠️ A diferença entre treino e teste sugere possível *overfitting*, porém moderado, pois o desempenho no teste continua elevado.

Validação Cruzada

- Para garantir a robustez da avaliação do modelo e reduzir a variância na estimativa de erro, foi aplicada uma validação cruzada com 5 *folds.*
- Os scores obtidos em cada *fold* foram: [0.9131, 0.9242, 0.9116, 0.9178, 0.9175].
- A **Média CV** (Validação Cruzada) foi de **0.9168**, com **desvio padrão de 0.0044.**

Esse desvio-padrão maior em comparação a treinamentos sem SMOTE indica que o desempenho do modelo é um pouco mais variável entre os subconjuntos dos dados, o que é esperado ao lidar com dados balanceados artificialmente.

Curva de Aprendizado

A análise da curva de aprendizado (que avalia a evolução do desempenho do modelo em função do volume de dados de treino) demonstrou que:

- **Com 8% das amostras de treino (1610 amostras): treino = 1 e validação = 0.9226**
  - O modelo obteve 100% de acerto no treino e 92,26% na validação, indicando bom ajuste aos dados de treino, mas houve uma leve queda de desempenho ao generalizar para dados novos.
- **Com 80% das amostras de treino (16104 amostras): treino = 1 e validação = 0.9163**
  - Ao aumentar para 80% das amostras, o modelo manteve 100% de acerto no treino, mas a acurácia na validação teve uma leve queda para 91,63%.

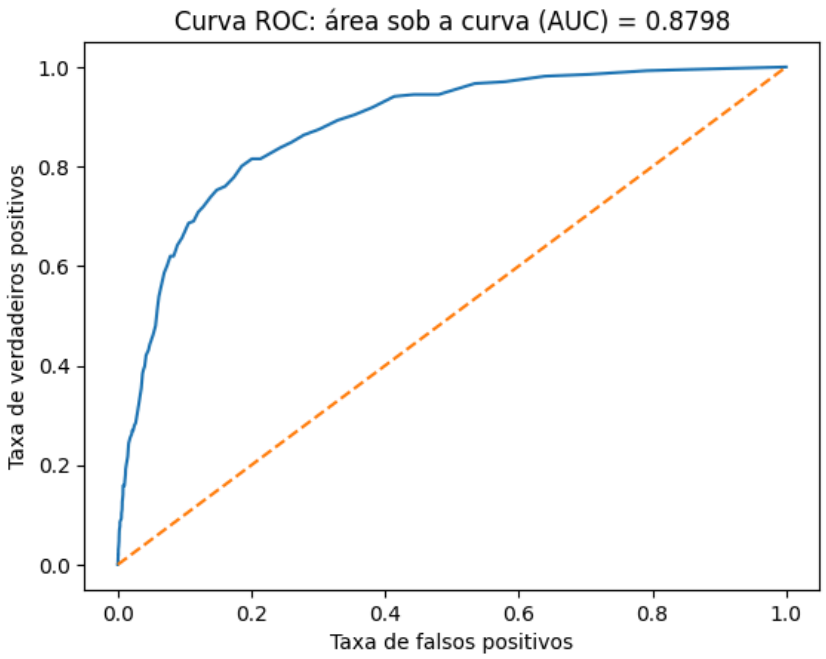
- Isso mostra que, apesar de usar muito mais dados, o modelo não melhorou na capacidade de prever dados novos, provavelmente devido ao foco em aprender os detalhes dos dados de treino.

Apesar da acurácia de treino perfeita ao longo de todo o processo, a validação apresenta uma leve queda à medida que mais dados são usados, sinalizando que o modelo pode estar limitado pela complexidade dos dados e pelo ajuste do balanceamento feito pelo SMOTE.

Matriz de Confusão e Relatório de Classificação

Esse resultado mostra que o modelo tem boa performance para a classe majoritária (False), com alta precisão (0.95) e recall (0.98), mas desempenho menor para a classe minoritária (True), com baixa precisão (0.46) e recall (0.27), indicando dificuldade em identificar corretamente os casos positivos, apesar da acurácia geral elevada (0.93).

Curva ROC e área sob a curva (AUC)



O modelo, ao ser balanceado com a técnica de SMOTE, alcançou o desempenho evidenciado pela área sob a curva (AUC) de **0.8798**. Este resultado demonstra uma boa capacidade em diferenciar corretamente as classes, superando o desafio comum do desbalanceamento de dados.

Estatística de Kolmogorov-Smirnov (KS)

O modelo apresentou um valor de KS (Kolmogorov-Smirnov) de **0.6154**, indicando boa capacidade discriminativa entre clientes adimplentes e inadimplentes. Esse resultado está acima do patamar de 0,6, geralmente considerado referência de alta performance em modelos de crédito.

🧰 Tecnologias Utilizadas

O projeto foi integralmente desenvolvido em Python, aderindo às recomendações, utilizando as bibliotecas abaixo:

- **Pandas**: manipulação e análise dos DataFrames.
- **NumPy**: operações numéricas.
- **Seaborn** e **Matplotlib**: para visualização de dados para análise das distribuições e identificação de outliers.
- **Scikit-Learn**: para modelagem - Pipeline, instanciamento do modelo e métricas de avaliação.
- **Imbalanced-learn:** para lidar com classes desbalanceadas, aplicando SMOTE para gerar dados sintéticos da classe minoritária corretamente durante o treinamento do modelo.

📌 Conclusões e Recomendações Finais

O projeto implementou um pipeline de ponta a ponta, abrangendo desde a limpeza de dados até a predição. Este sistema, com os devidos ajustes, pode ser aplicado a outros cenários de negócio, como a previsão de Churn, um desafio persistente em diversas empresas.

Com o objetivo principal atingido, espera-se que haja a oferta de **subsídio para a tomada de decisão envolvendo concessão de crédito**, ao ajudar a identificar cenários com maior probabilidade de atraso ou não pagamento.

O modelo Random Forest Classifier e balanceado com a técnica SMOTE, demonstrou um desempenho notável no cenário proposto. Embora tenha atingido uma acurácia de treino perfeita (1.0), com uma ligeira redução na validação com um volume maior de dados, o modelo manteve uma **forte capacidade de generalização** para dados não observados.

Foi atingida uma AUC (área sob a curva ROC) de 0.8798 e um KS de 0.6154, este último superando o limite de 0.4, referência para alta performance em modelos de crédito. Isso indica uma boa capacidade de distinguir entre clientes adimplentes e inadimplentes, mesmo com a diferença entre a acurácia de treino e teste (0.9295) sugerindo um possível *overfitting*.

Como próximos passos, recomenda-se que o modelo seja **avaliado periodicamente**, com novos treinamentos realizados à medida que novos dados se tornem disponíveis, para manter e, se possível, elevar o desempenho.

📚 Referências Bibliográficas

Thomas, L. C., Edelman, D. B., & Crook, J. N. (2002). *Credit Scoring and Its Applications*. SIAM – Society for Industrial and Applied Mathematics.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

SILVA, André. *Como usar Pipelines no Scikit-Learn*. **Data Hackers**, 7 dez. 2020. Disponível em: https://medium.com/data-hackers/como-usar-pipelines-no-scikit-learn-1398a4cc6ae9. Acesso em: jul. 2025.

NAVLANI, A. Understanding Random Forests Classifier in Python. **DataCamp**, 2018. Disponível em: https://www.datacamp.com/pt/tutorial/random-forests-classifier-python. Acesso em: jul. 2025.

FILHO, Mario. **Random Forest na Prática (Scikit-learn / Python)**. YouTube, 2020. Disponível em: https://www.youtube.com/watch?v=RtA1rjhuavs. Acesso em: jul. 2025.

Anexo I - Tratamento das Bases

🗃️ base_cadastral

ID_CLIENTE

- sem valores nulos ou outras inconsistências

DATA_CADASTRO

- sem valores nulos ou outras inconsistências
- conversão para o formato de data
- não foram identificados outliers nas análises

CEP_2_DIG

- sem valores nulos e, na análise gráfica, não foram identificados outliers

DDD

- na análise de valores fora do padrão constituído por 2 algarismos decimais, que representam os 2 primeiros dígitos do DDD, foram identificados alguns iniciando com **(**
  - ➢ ex: (0, (1, (2, (3 …
- também haviam valores nulos
- nesses dois cenários, os valores foram substituídos por DDDs estimados
  - ➢ essa estimativa se baseou no CEP, através de um dicionário no seguinte formato: {'2 primeiros dígitos do CEP': 'DDD correspondente estimado'}
- não foram identificados outliers nas análises

Estes, junto com os registros nulos, foram substituídos por DDDs.

PORTE, **DOMINIO_EMAIL**, **SEGMENTO_INDUSTRIAL**, **FLAG_PF**

- valores nulos substituídos por “NÃO INFORMADO”

Coluna criada: TEMPO_CADASTRO_ANOS

- representa a "idade" do cliente, com base na data de referência de Novembro/2021, que é a data mais recente da base_pagamentos_teste
- é calculada pela diferença entre Novembro/2021 e a **DATA_CADASTRO**
- essa criação ocorreu pois a é uma informação que pode ser relevante para o treinamento do modelo

🗃️ base_info

ID_CLIENTE

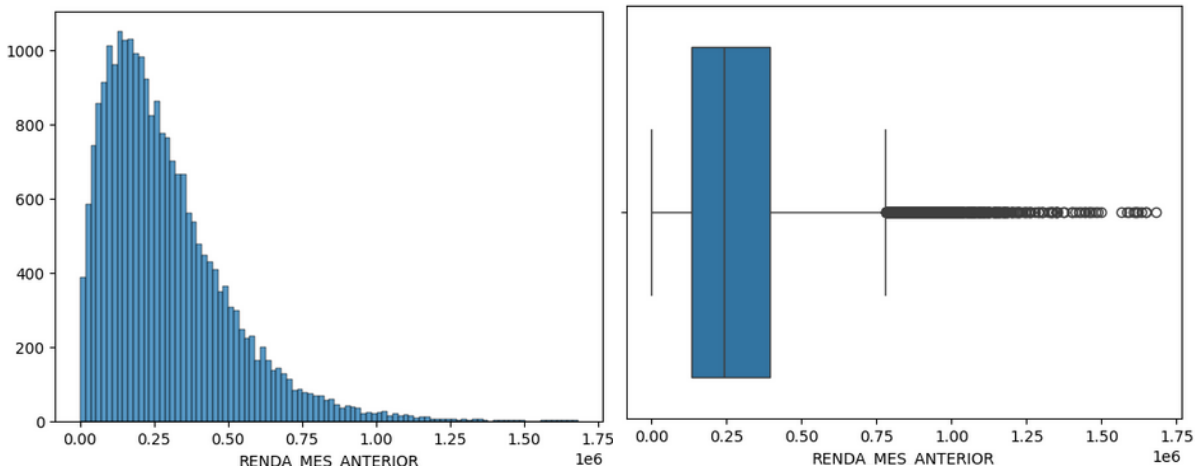
- sem valores nulos ou outras inconsistências

SAFRA_REF

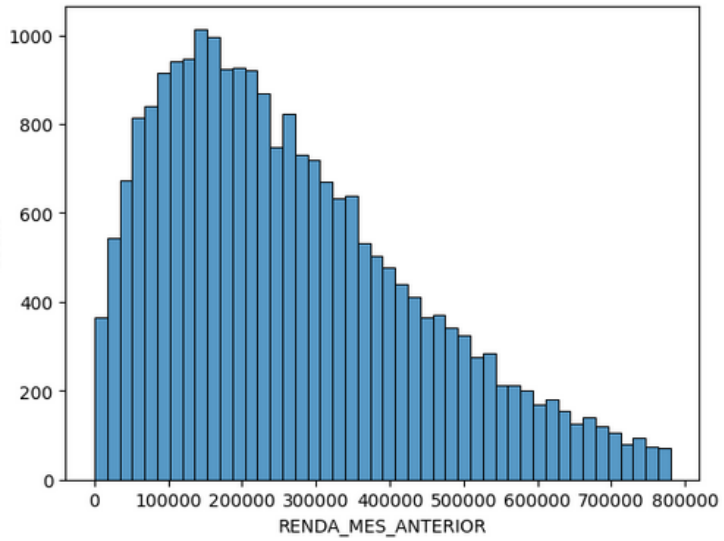
- sem valores nulos ou outras inconsistências
- continham somente o mês e ano
- para manter a mesma granularidade das outras tabelas e colunas, foi adicionado o dia 01
- conversão para o formato de data
- não foram identificados outliers

RENDA_MES_ANTERIOR

- a análise gráfica indicou outliers à direita da distribuição:



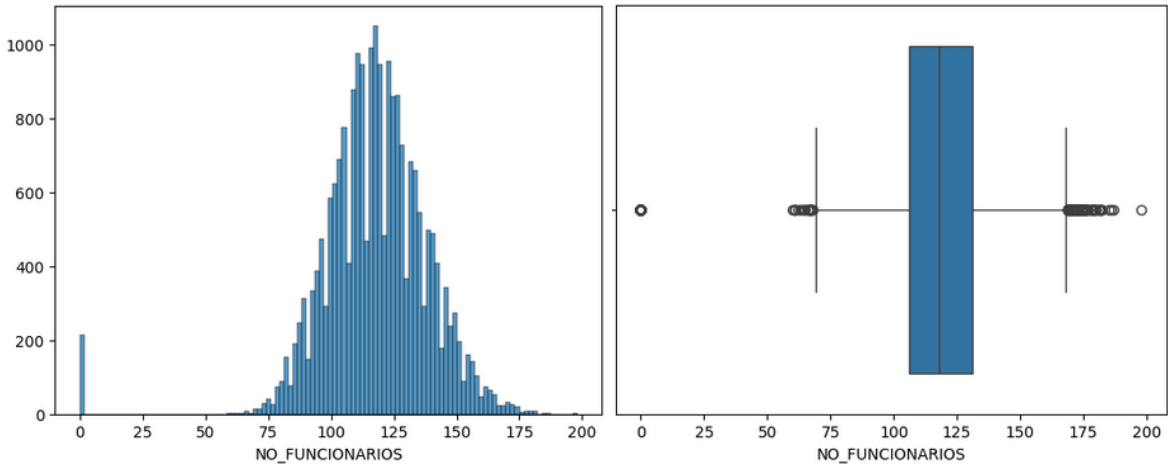
- trata-se de uma distribuição assimétrica à direita com comportamento exponencial
- para isso, foi utilizado o Método de Interseção Interquartil (IQR) para remover os 761 valores acima do limite superior, reduzindo o dataframe em 3,1%, que é um valor aceitável
- abaixo, o mesmo histograma após remoção dos outliers



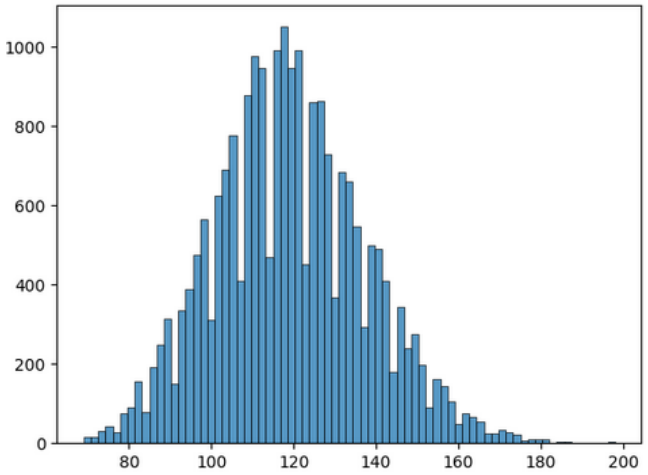
- valores nulos preenchidos com a mediana, após remoção dos outliers

NO_FUNCIONARIOS

- a análise gráfica indicou outliers à esquerda da distribuição:



  - trata-se de uma distribuição próxima à Gaussiana, com um deslocamento à direita
  - para isso, foi utilizado o IQR para remover os 234 valores abaixo do limite inferior
  - abaixo, o mesmo histograma após remoção dos outliers



  - valores nulos preenchidos com a mediana, após remoção dos outliers
  - também foram removidos os valores iguais a zero
  - conversão do formato Float para Inteiro

🗃️ base_pagamentos_desenvolvimento

ID_CLIENTE, **TAXA**

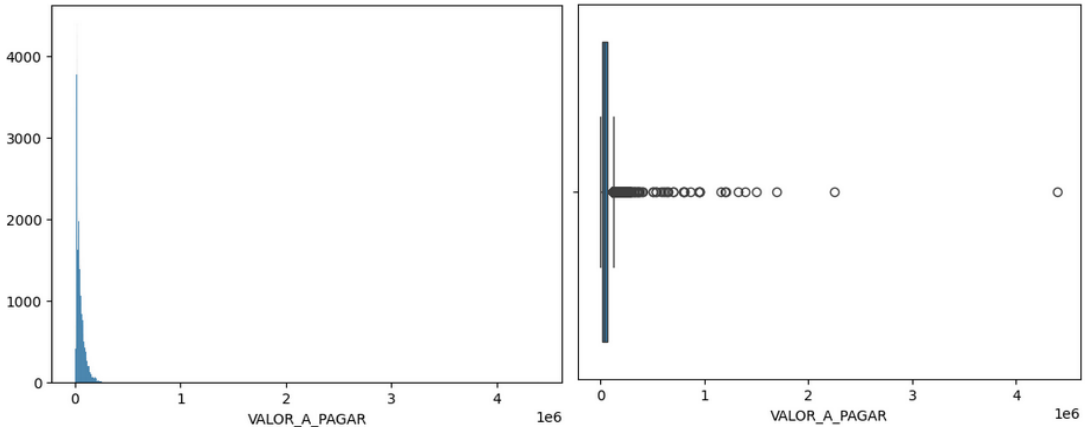
- sem valores nulos ou outras inconsistências

SAFRA_REF, **DATA_EMISSAO_DOCUMENTO**, **DATA_PAGAMENTO**

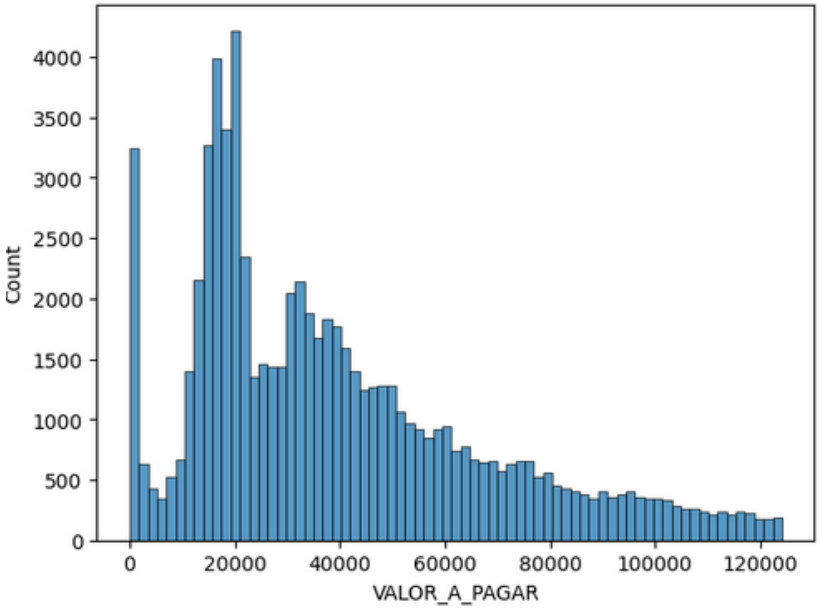
- sem valores nulos ou outras inconsistências
- conversão para data

VALOR_A_PAGAR

- a análise gráfica indicou outliers à direita da distribuição:



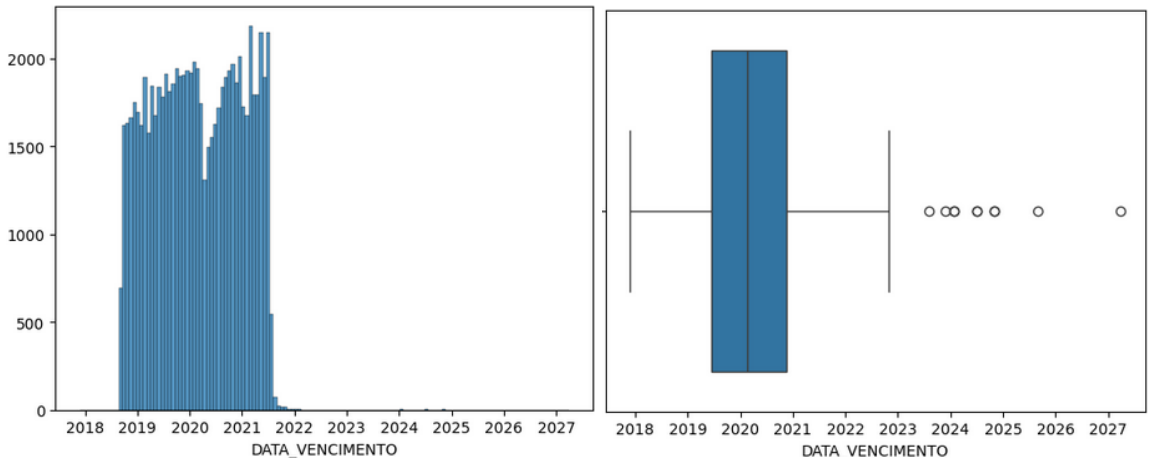
- há valores à direita que alongam a calda
- também foi usado o IQR para remover os valores à direita
- abaixo, a mesma distribuição após remoção dos outliers



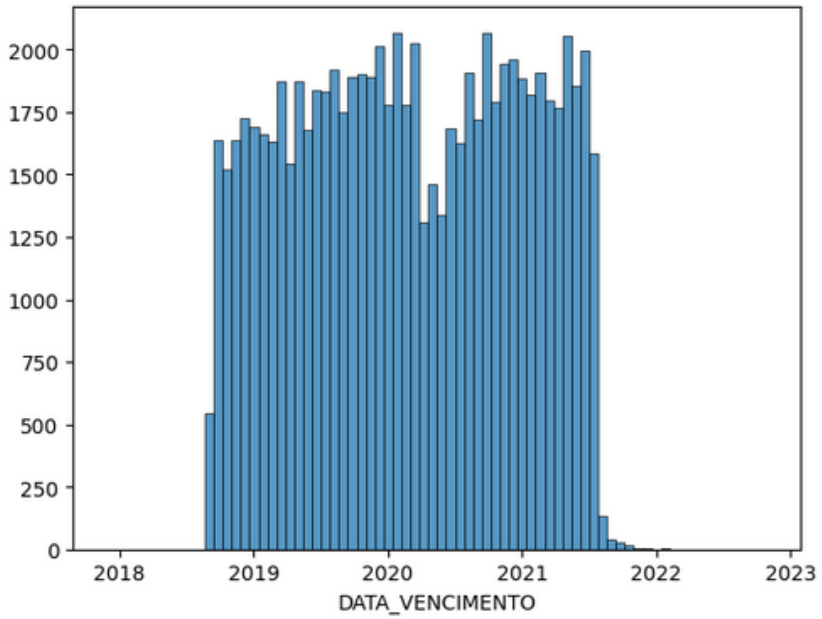
- valores nulos preenchidos com a mediana, após remoção dos outliers
- também foram removidos os valores iguais a zero

DATA_VENCIMENTO

- sem valores nulos
- a análise gráfica indicou outliers à direita da distribuição:



- valores acima do limite superior
- também foi usado o IQR para remover os valores à direita
- abaixo, a mesma distribuição após remoção dos outliers



- valores nulos preenchidos com a mediana, após remoção dos outliers
- também foram removidos os valores iguais a zero

🧰 base_pagamentos_teste

ID_CLIENTE

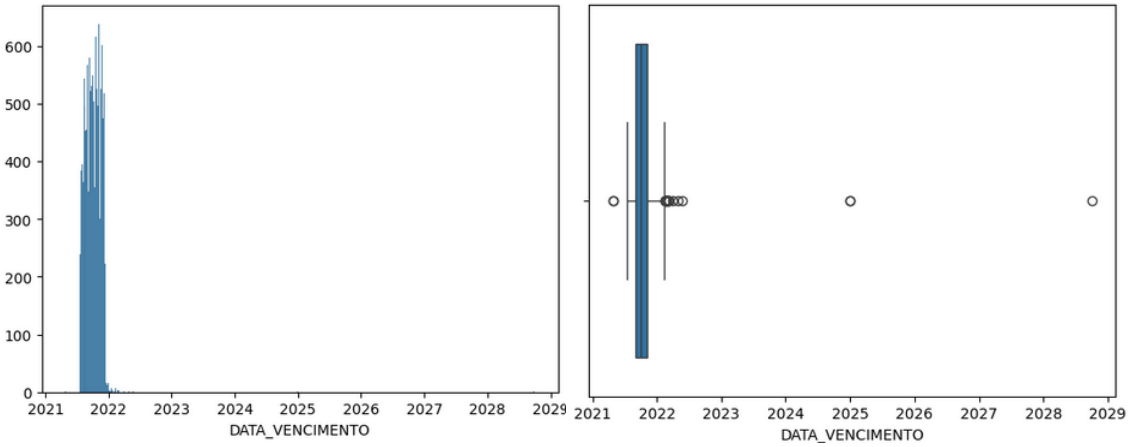
- sem valores nulos ou outras inconsistências

SAFRA_REF, DATA_EMISSAO_DOCUMENTO

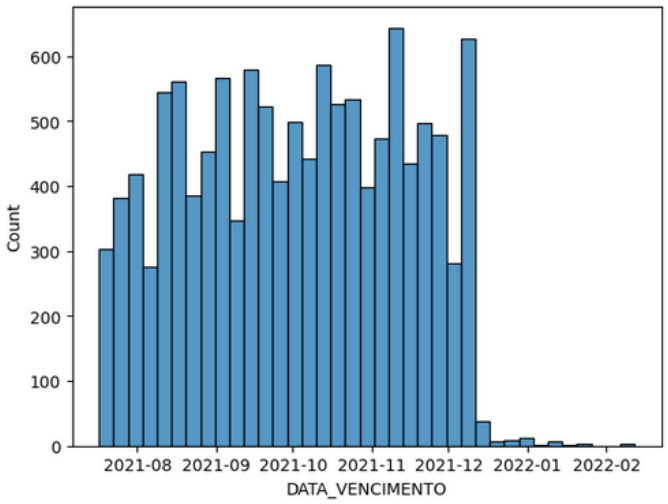
- sem valores nulos ou outras inconsistências
- conversão para data

DATA_VENCIMENTO

- sem valores nulos
- a análise gráfica indicou outliers à direita e à esquerda da distribuição:



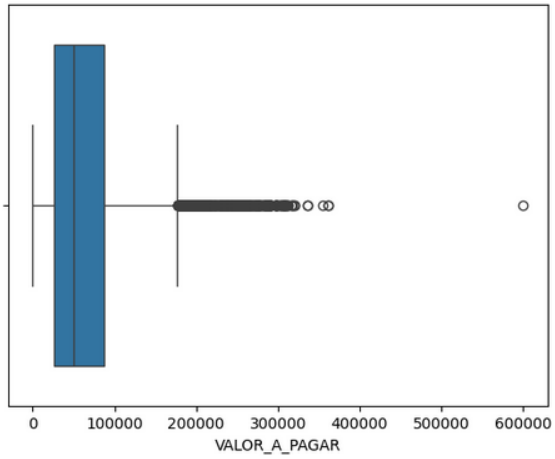
- também foi usado o IQR para remover os valores à direita e à esquerda
- abaixo, a mesma distribuição após remoção dos outliers



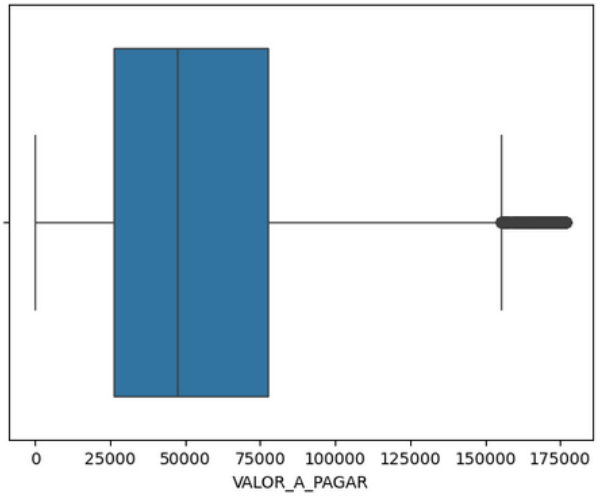
- conversão para data

VALOR_A_PAGAR

- a análise gráfica indicou outliers à direita da distribuição:



- também foi usado o IQR para remover os valores à direita
- abaixo, a mesma distribuição após remoção dos outliers



- preenchimento dos nulos com a mediana
- conversão para data

TAXA

- sem valores nulos ou outras inconsistências

✳ Observações

1️⃣ Em todos os cenários de outliers, foi escolhido o Método de Interseção Interquartil pois ele funciona bem para **distribuições normais ou homogêneas**, que foram os casos encontrados. Isso se dá pelo fato desse método usar os Quartis, que são menos influenciados por valores extremos. Assim, há uma remoção eficiente de outliers sem distorcer a estrutura dos dados.

2️⃣ Os valores nulos das variáveis numéricas foram preenchidos pela mediana pois por ser uma medida de tendência central robusta, também menos sensível a outliers e assimetrias. Isso assegura que o preenchimento não afete significativamente a distribuição original da variável, mantendo a integridade estatística dos dados.