

# Documentação projeto 03 laboratoria

## CONTEXTO

### Caso

No contexto recente, a diminuição das taxas de juros no mercado desencadeou um aumento significativo na demanda por solicitações de crédito. Os clientes veem uma oportunidade favorável para financiar compras importantes ou consolidar dívidas existentes, o que levou a uma afluência de solicitações de empréstimo no banco "Super Caja". A equipe de análise de crédito do banco está enfrentando uma carga de trabalho avassaladora devido à análise manual necessária para cada solicitação de empréstimo de clientes individuais. Essa metodologia manual resultou em um processo ineficiente e demorado, afetando negativamente a eficácia e a rapidez com que as solicitações de empréstimo são processadas. A situação se torna mais crítica devido à crescente preocupação com a taxa de inadimplência, um problema que está afetando cada vez mais a indústria financeira, aumentando a pressão sobre os bancos para identificar e mitigar os riscos associados ao crédito.

Com o objetivo de enfrentar esse desafio, a proposta é a automação do processo de análise utilizando técnicas avançadas de análise de dados, visando melhorar a eficiência, precisão e rapidez na avaliação das solicitações de crédito. Além disso, o banco já possui uma métrica para identificar clientes com pagamentos em atraso, o que poderia ser uma ferramenta valiosa para integrar na classificação de risco dentro do novo sistema automatizado.

### Objetivo:

O objetivo da análise é desenvolver um score de crédito a partir de uma análise de dados e avaliação do risco relativo que possa classificar os solicitantes em diferentes categorias de risco com base em sua probabilidade de inadimplência. Essa classificação permitirá ao banco tomar decisões informadas sobre quem conceder crédito, reduzindo assim o risco de empréstimos não reembolsáveis. Além disso, a integração da métrica existente de pagamentos em atraso fortalecerá a capacidade do modelo de identificar riscos, contribuindo assim para a solidez financeira e eficiência operacional do banco.

### Ferramentas e Tecnologias:

#### 1. Ferramentas, linguagens e insumos

## 1.1 Ferramentas e/ou plataformas

Neste projeto você usará quatro ferramentas do Google, uma para manipulação de dados em SQL, um para criar apresentações, uma para visualizar o resultado do seu trabalho e a última para manipulação de dados em Python:

- Google BigQuery: Data warehouse que permite o processamento de grandes volumes de dados.
- Google Colab: Plataforma para trabalhar com a linguagem de programação Python em Notebooks.
- Apresentações Google: ferramenta para criação e edição de apresentações.
- Google Looker Studio: ferramenta para criação e edição de painéis e relatórios de dados.

## 1.2 Linguagens

Você usará a linguagem SQL no BigQuery e Python no Google Colab.

## 1.3 Insumos

Este conjunto de dados contém dados sobre empréstimos concedidos a um grupo de clientes do banco. Os dados estão divididos em 4 tabelas, a primeira com dados do usuário/cliente, a segunda com dados do tipo empréstimo, a terceira com o comportamento de pagamento desses empréstimos, e a quarta com a informação dos clientes já identificados como inadimplentes.

O conjunto de dados está disponível para download neste link [conjunto de dados](#). Observe que é um arquivo compactado, você terá que descompactá-lo para acessar os arquivos com as tabelas.

Abaixo, você pode consultar a descrição das variáveis que compõem as tabelas deste conjunto de dados:

Arquivo	Variável	Descrição
user_info	user id	Número de identificação do cliente (único para cada cliente)
	age	Idade do cliente
	sex	Gênero do cliente
	last month salary	Último salário mensal que o cliente informou ao banco
	number dependents	Número de dependentes
loans_outstanding	loan id	Número de identificação do empréstimo (único para cada empréstimo)
	user id	Número de identificação do cliente
	loan type	Tipo de empréstimo (real estate = imóveis, others= outros)
loans_detail	user id	Número de identificação do cliente
	more 90 days overdue	Número de vezes que o cliente apresentou atraso superior a 90 dias
	using lines not secured personal assets	Quanto o cliente está utilizando em relação ao seu limite de crédito, em linhas que não são garantidas por bens pessoais, como imóveis e automóveis
	number times delayed payment loan 30 59 days	Número de vezes que o cliente atrasou o pagamento de um empréstimo (entre 30 e 59 dias)

Arquivo	Variável	Descrição
	debt ratio	Relação entre dívidas e ativos do cliente. Taxa de endividamento = Dívidas / Patrimônio
	number times delayed payment loan 60 89 days	Número de vezes que o cliente atrasou o pagamento de um empréstimo (entre 60 e 89 dias)
default	user id	Número de identificação do cliente
	default flag	Classificação dos clientes inadimplentes (1 para clientes já registrados alguma vez como inadimplentes, 0 para clientes sem histórico de inadimplência)

## 2.1 Processar e preparar a base de dados



### 2.1.1 Conectar/importar dados para ferramentas

Para iniciar o projeto, os arquivos .zip do conjunto de dados foram descompactados, carregados para o Google Drive e convertidos para o formato de Planilhas Google. Em seguida, as bases de dados foram importadas como tabelas para o Google Cloud BigQuery.



## 2.1.2 Identificar e tratar valores nulos

Encontrei um problema de tipagem na linha 4931

The screenshot shows the Google Cloud BigQuery interface. On the left, the 'Explorer' sidebar lists projects, datasets, and tables, including 'my-project-laboratoria.dadoslaboratorioproject03.user\_info'. In the main area, a query titled 'Consulta sem título' is being run:

```
1 # Verificando se há valores nulos em todas as variáveis na tabela user_info:
2 SELECT
3     COUNT(*) AS total_linhas,
4     COUNTIF(user_id IS NULL) AS user_id_nulos,
5     COUNTIF(age IS NULL) AS age_nulos,
6     COUNTIF(sex IS NULL) AS sex_nulos,
7     COUNTIF(CAST(last_month_salary AS FLOAT64) IS NULL) AS last_month_salary_nulos,
8     COUNTIF(number_dependents IS NULL) AS number_dependents_nulos
9 FROM
10 `my-project-laboratoria.dadoslaboratorioproject03.user_info`
```

The results show an error message: 'Error while reading table: my-project-laboratoria.dadoslaboratorioproject03.user\_info, error message: Could not convert value to integer. Row 4931; Col 3. File: 1SnZWRZ3Dw1ks90yZLUGt0CkCqj57K858bpN4icW6N0l'.

Usei o cast na tentativa que o tipo float resolvesse e não obtive sucesso

The screenshot shows the Google Cloud BigQuery interface again. The same query is run, but this time it includes a cast operation on the salary column:

```
1 # Verificando se há valores nulos em todas as variáveis na tabela user_info:
2 SELECT
3     COUNT(*) AS total_linhas,
4     COUNTIF(user_id IS NULL) AS user_id_nulos,
5     COUNTIF(age IS NULL) AS age_nulos,
6     COUNTIF(sex IS NULL) AS sex_nulos,
7     COUNTIF(CAST(last_month_salary AS FLOAT64) IS NULL) AS last_month_salary_nulos,
8     COUNTIF(number_dependents IS NULL) AS number_dependents_nulos
9 FROM
10 `my-project-laboratoria.dadoslaboratorioproject03.user_info`
```

The results still show the same error message: 'Error while reading table: my-project-laboratoria.dadoslaboratorioproject03.user\_info, error message: Could not convert value to integer. Row 4931; Col 3. File: 1SnZWRZ3Dw1ks90yZLUGt0CkCqj57K858bpN4icW6N0l'.

Como a coluna de salario já estava apresentando erro de leitura nesta linha 4931, já calculei a mediana de salario usando Idade e sexo como critérios e para possivelmente fazer a substituição.

The screenshot shows a Google Sheets document titled 'user\_info'. The table contains 26 rows of data. The columns are labeled as follows: A (user\_id), B (age), C (sex), D (last\_month\_salary), E (number\_dependents), F (ZFORMS), G (age), H (sex), I (MEDIA SALARIAL), and J (MEDIANA SALARIO). The 'last\_month\_salary' column (D) is highlighted in yellow. The data includes various user IDs, ages (e.g., 55F, 44M, 54M, 54F, 42M, 83F, 78M, 55M, 41M, 49F, 58M, 77M, 68F, 41M, 64M, 69F, 78M, 73F, 51F, 42M, 65F, 75M, 41M, 53F, 59M), and their respective salaries and dependents.

Verifiquei que subindo o csv a coluna de last\_month\_salary é tipada como float, portanto resolvi criar o schema manualmente desta tabela.

The screenshot shows the Google Cloud BigQuery interface. On the left, there is an 'Explorer' sidebar with various project and dataset options. In the main area, a table named 'user\_info02' is displayed. The schema tab is selected, showing the following columns:

Nome do campo	Tipo	Modo	Chave	Compilação	Valor padrão	Tags de políticas	Políticas de dados	Descrição
user_id	INTEGER	NULABLE	-	-	-	-	-	-
age	INTEGER	NULABLE	-	-	-	-	-	-
sex	STRING	NULABLE	-	-	-	-	-	-
last_month_salary	FLOAT	NULABLE	-	-	-	-	-	-
number_dependents	INTEGER	NULABLE	-	-	-	-	-	-

e o erro foi corrigido:

```

1 SELECT
2   COUNT(*) AS total_linhas,
3   COUNTIF(user_id IS NULL) AS user_id_nulos,
4   COUNTIF(age IS NULL) AS age_nulos,
5   COUNTIF(sex IS NULL) AS sex_nulos,
6   COUNTIF(SAFE_CAST(last_month_salary AS FLOAT64) IS NULL) AS last_month_salary_nulos,
7   COUNTIF(number_dependents IS NULL) AS number_dependents_nulos
8 FROM
9   `my-project-laboratoria.dadoslaboratorioproject03.user_info`
10
11
12
13
14
15

```

**Consulta concluída**

**Resultados da consulta**

Linha	totalLinhas	user_id_nulos	age_nulos	sex_nulos	last_month_salary_nulos	number_dependents_nulos
1	36000	0	0	0	7199	943

--Resultados da tabela user\_info: o total de dados é de 36.000 mil, foram identificados 7.199 valores nulos na variável last\_month\_salary e 943 valores nulos na variável number\_dependents

na tabela loans\_detail:

```

16   COUNT(*) AS total_linhas,
17   COUNTIF(user_id IS NULL) AS user_id_nulos,
18   COUNTIF(more_90_days_overdue IS NULL) AS more_90_days_overdue_nulos,
19   COUNTIF(using_lines_not_secured_personal_assets IS NULL) AS using_lines_not_secured_personal_assets_nulos,
20   COUNTIF(number_times_delayed_payment_loan_30_59_days IS NULL) AS number_times_delayed_payment_loan_30_59_days_nulos,
21   COUNTIF(debt_ratio IS NULL) AS debt_ratio_nulos,
22   COUNTIF(number_times_delayed_payment_loan_60_89_days IS NULL) AS number_times_delayed_payment_loan_60_89_days_nulos
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```

**Consulta concluída**

**Resultados da consulta**

Linha	totalLinhas	user_id_nulos	more_90_days_overdue_nulos	using_lines_not_secured_personal_assets_nulos	number_times_delayed_payment_loan_30_59_days_nulos	debt_ratio_nulos	number_times_delayed_payment_loan_60_89_days_nulos
1	36000	0	0	0	0	0	0

--Resultados da consulta: o total de dados é de 36.000 mil, não foram identificados valores nulos na tabela loans\_outstanding

```

23 COUNT(*) AS total_linhas;
24 COUNTIF(user_id IS NULL) AS user_id_nulos;
25 COUNTIF(loan_id IS NULL) AS more_30_days_overdue_nulos;
26 COUNTIF(using_lines IS NULL) AS more_30_days_overdue_nulos;
27 COUNTIF(number_times_delayed_payment_loan_30_59_days IS NULL) AS number_times_delayed_payment_loan_30_59_days_nulos;
28 COUNTIF(debt_ratio IS NULL) AS debt_ratio_nulos;
29 COUNTIF(number_times_delayed_payment_loan_60_89_days IS NULL) AS number_times_delayed_payment_loan_60_89_days_nulos;
30 COUNTIF(number_times_delayed_payment_loan_90_109_days IS NULL) AS number_times_delayed_payment_loan_90_109_days_nulos;
31 FROM `my-project-laboratoria.dadoslaboratoriaproject03.loans_outstanding`;
32
33 # Verificando se há valores nulos em todas as variáveis na tabela loans_outstanding
34 SELECT
35     COUNT(*) AS total_linhas,
36     COUNTIF(loan_id IS NULL) AS loan_id_nulos,
37     COUNTIF(user_id IS NULL) AS user_id_nulos,
38     COUNTIF(loan_type IS NULL) AS loan_type_nulos
39 FROM `my-project-laboratoria.dadoslaboratoriaproject03.loans_outstanding`;
40
41

```

**Resultados da consulta**

Linha	total_linhas	loan_id_nulos	user_id_nulos	loan_type_nulos
1	305335	0	0	0

Resultados por página: 50 ▾ 1 – 1 de 1 | < > >>

-- Resultados da consulta: o total de dados é de 305.335 mil, não foram identificados valores nulos



### 2.1.3 🔍 Identificar e tratar valores duplicados

```

1 #consulta para identificar valores duplicados na tabela loans_outstanding
2 SELECT
3     loan_id,
4     user_id,
5     loan_type,
6     COUNT(*) AS num_duplicates
7 FROM
8     `my-project-laboratoria.dadoslaboratoriaproject03.loans_outstanding`
9 GROUP BY
10     loan_id, user_id, loan_type
11 HAVING
12     COUNT(*) > 1

```

**Resultados da consulta**

Informações do job **Resultados** Visualização JSON Detalhes da execução Gráfico de execução

● Não há dados para exibir.

Resultados por página: 50 ▾ 1 – 0 de 0 | < > >>

-- Resultado da consulta: Não constam valores duplicados

```

    SELECT
        user_id,
        age,
        sex,
        last_month_salary,
        number_dependents,
        COUNT(*) AS num_duplicatas
    FROM `my-project-laboratoria.dadoslaboratoriaproject03.user_info`
    GROUP BY
        user_id,
        age,
        sex,
        last_month_salary,
        number_dependents
    HAVING
        COUNT(*) > 1;

```

The screenshot shows the Google Cloud BigQuery interface. On the left, the Explorer sidebar lists various datasets and tables, including 'user\_info', 'loans\_detail', 'loans\_outstanding', and 'user\_info'. The main area displays a query titled 'Consulta sem título' (Untitled Query) with the code provided above. Below the query, the results section shows a single row of data:

Linha	corr_60_89	corr_30_59	correlacao	correlacao_1
1	0.992175526340...	0.982916806614...	0.986553645498...	0.015012138743...

-- Resultado da consulta: Não constam valores duplicados



### 2.1.4 Identificar e gerenciar dados fora do escopo da análise

```

    CORR(
        SAFE_CAST(more_90_days_overdue AS FLOAT64),
        SAFE_CAST(number_times_delayed_payment_loan_60_89_days AS FLOAT64)
    ) AS corr_60_89,
    CORR(
        SAFE_CAST(more_90_days_overdue AS FLOAT64),
        SAFE_CAST(number_times_delayed_payment_loan_30_59_days AS FLOAT64)
    ) AS corr_30_59,
    CORR(
        number_times_delayed_payment_loan_30_59_days,
        number_times_delayed_payment_loan_60_89_days
    ) AS correlacao,
    CORR(
        SAFE_CAST(using_lines_not_secured_personal_assets AS FLOAT64),
        SAFE_CAST(debt_ratio AS FLOAT64)
    ) AS correlacao
    FROM `my-project-laboratoria.dadoslaboratoriaproject03.loans_detaila_teste`;

```

The screenshot shows the Google Cloud BigQuery interface. On the left, the Explorer sidebar lists various datasets and tables, including 'Correlação\_project03'. The main area displays a query titled 'Correlação\_project03' with the code provided above. Below the query, the results section shows a single row of data:

Linha	corr_60_89	corr_30_59	correlacao	correlacao_1
1	0.992175526340...	0.982916806614...	0.986553645498...	0.015012138743...

The screenshot shows the Google Cloud BigQuery interface. The left sidebar has an 'Explorer' section with a tree view of datasets and tables. The main area contains a query editor with the following SQL code:

```

#verificando desvio padrão
SELECT
    more_59,
    more_90,
    dias_89
FROM
    STDEEV_SANT(more_90.days.override) AS more_90,
    STDEEV_SANT(number_times.delayed_payment_loan_00_89.days) AS dias_89
    FROM
        `my-project-laboratoria.dadoslaboratoriaproject03.loans_detail`

```

The results table shows one row of data:

Linha	more_59	more_90	dias_89
1	4.144020438225...	4.121364668426...	4.105514755101...

fez a junção de defalt com user\_info para validar as correlações e desvios padrão.

The screenshot shows the Google Cloud BigQuery interface. The left sidebar has an 'Explorer' section with a tree view of datasets and tables. The main area contains a query editor with the following SQL code:

```

Junção user_info & defalt
SELECT
    d.* -- Todas as colunas da tabela defalt;
    ui.* EXCEPT(user_id) -- Todas as colunas da tabela user_info exceto user_id
FROM
    `my-project-laboratoria.dadoslaboratoriaproject03.defalt` AS d
LEFT JOIN
    `my-project-laboratoria.dadoslaboratoriaproject03.user_info` AS ui
ON
    d.user_id = ui.user_id

```

The results table shows three rows of data:

Linha	user_id	default_flag	age	sex	last_month_salary	number_dependents
1	49	0	83	M	4694.0	0
2	53	0	84	F	3001.0	1
3	62	0	88	M	null	null

voltou pra corrigir os nulos usando mediana de las\_month\_salary em user info já que vou precisar da junção dela com details para validar correlação e desvio padrão

```

3 COUNTIF(user_id IS NULL) AS user_id_nulos,
4 COUNTIF(age IS NULL) AS age_nulos,
5 COUNTIF(last_month_salary IS NULL) AS last_month_salary_nulos,
6 COUNTIF(CAST(last_month.salary AS FLOAT64) IS NULL) AS last_month_salary_nulos,
7 COUNTIF(number_dependents IS NULL) AS number_dependents_nulos,
8 FROM `my-project-laboratoria.dadoslaboratoriaproject03.user_info`
9
10 #TRATAMENTO DOS NULOS NA TABELA user_info E CRIANDO A VIEW JA CORRIGIDA
11 CREATE OR REPLACE VIEW `my-project-laboratoria.dadoslaboratoriaproject03.user_info_corrigida` AS
12
13 SELECT
14   user_id,
15   age,
16   sex,
17   IFNULL(last_month_salary, (SELECT APPROX_QUANTILES(last_month_salary, 2)[OFFSET(1)]) FROM `my-project-laboratoria.dadoslaboratoriaproject03.user_info` ) AS last_month_salary,
18   IFNULL(number_dependents, (SELECT APPROX_QUANTILES(number_dependents, 2)[OFFSET(1)]) FROM `my-project-laboratoria.dadoslaboratoriaproject03.user_info` ) AS number_dependents
19
20 # Verificando se há valores nulos em todas as variáveis na tabela loans_detail:
21
22 SELECT
23   COUNT(*) AS total_linhas,
24   COUNT(*) AS total_nulos
25
26 Consulta concluída

```

Resultados da consulta

Informações do job	Resultados	Detalhes da execução	Gráfico de execução
--------------------	------------	----------------------	---------------------

criei a view desta junção

```

1 CREATE OR REPLACE VIEW `my-project-laboratoria.dadoslaboratoriaproject03.default_user_info` AS
2
3 SELECT
4   d.* -- Todas as colunas da tabela default
5   ui.* EXCEPT(user_id) -- Todas as colunas da tabela user_info exceto user_id
6   FROM `my-project-laboratoria.dadoslaboratoriaproject03.default` AS d
7   LEFT JOIN `my-project-laboratoria.dadoslaboratoriaproject03.user_info_corrigida` AS ui
8   ON d.user_id = ui.user_id
9
10 Esta consulta vai processar 0 B quando executada.

```

Resultados da consulta

Informações do job	Resultados	Visualização	JSON	Detalhes da execução	Gráfico de execução	
Linha	user_id	default_flag	age	sex	last_month_salary	number_dependents
1	49	0	63	M	4694.0	0
2	53	0	84	F	3001.0	1
3	62	0	88	M	5416.0	0

```

1 SELECT
2   CORR(default_flag, last_month_salary) AS correlacao,
3   CORR(default_flag, number_dependents) AS correlacao,
4   CORR(default_flag, age) AS correlacao
5
6 FROM
7   `my-project-laboratoria.dadoslaboratoriaproject03.default_user_info`
8
9 Esta consulta vai processar 0 B quando executada.

```

Resultados da consulta

Informações do job	Resultados	Visualização	JSON	Detalhes da execução	Gráfico de execução
Linha	correlacao	correlacao_1	correlacao_2		
1	-0.01957910230...	0.032595710531...	-0.07821770155...		



## 2.1.5 Identificar e tratar dados inconsistentes em variáveis categóricas

Na base de dados, não foi encontrado nenhum dado discrepante, encontrei somente algumas **inconsistências**. Notei que na tabela loans\_outstanding a variável loan\_type apresentava valores misturados em letras maiúsculas e minúsculas, sem nenhum padrão consistente. Para garantir a uniformidade dos dados, corri了 essas inconsistências, padronizando a nomenclatura e garantindo a separação adequada entre as palavras.

The screenshot shows the Google Cloud BigQuery interface. On the left, the 'Explorer' sidebar lists datasets and tables, including 'loans\_detail', 'loans\_outstanding', and 'loans\_outstanding.corrigida'. In the main area, a query editor window displays the following SQL code:

```
1 CREATE OR REPLACE VIEW `my-project-laboratoria.dadoslaboratorioproject03.loans_outstanding_corrigida` AS
2 SELECT
3     l.loan_id,
4     l.user_id,
5     UPPER(l.loan_type) AS loan_type,
6     FROM `my-project-laboratoria.dadoslaboratorioproject03.loans_outstanding` AS l
```

The status bar at the bottom indicates: 'Esta consulta vai processar 0 B quando executada.'



## 2.1.6 Identificar e tratar dados discrepantes em variáveis numéricas

Os quartis (Q1 e Q3) foram calculados para identificar e tratar outliers (valores discrepantes) nas variáveis numéricas. O tratamento foi realizado substituindo os valores fora dos limites ( $Q1 - 1.5 * IQR$  e  $Q3 + 1.5 * IQR$ ) pelos próprios limites.

**Obs: esta etapa s只 tratei depois da analise exploratária.**

```
CREATE OR REPLACE TABLE
`my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados` AS
WITH
base_outliers_tratados AS (
SELECT
    user_id,
    default_flag,
    age,
    faixa_etaria,
    last_month_salary,
    number_dependents,
    total_emprestimos,
    qtd_imoveis_emprestimos,
    qtd_outros_emprestimos,
    more_90_days_overdue,
    using_lines_not_secured_personal_assets,
    number_times_delayed_payment_loan_30_59_days,
    number_times_delayed_payment_loan_60_89_days,
```

```

debt_ratio,
-- Cálculo dos quartis para o tratamento de outliers
PERCENTILE_CONT(age, 0.25) OVER() AS age_q1,
PERCENTILE_CONT(age, 0.75) OVER() AS age_q3,
PERCENTILE_CONT(last_month_salary, 0.25) OVER() AS salary_q1,
PERCENTILE_CONT(last_month_salary, 0.75) OVER() AS salary_q3,
PERCENTILE_CONT(number_dependents, 0.25) OVER() AS dependents_q1,
PERCENTILE_CONT(number_dependents, 0.75) OVER() AS dependents_q3,
PERCENTILE_CONT(using_lines_not_secured_personal_assets, 0.25) OVER() AS using_lines_q1,
PERCENTILE_CONT(using_lines_not_secured_personal_assets, 0.75) OVER() AS using_lines_q3,
PERCENTILE_CONT(debt_ratio, 0.25) OVER() AS debt_ratio_q1,
PERCENTILE_CONT(debt_ratio, 0.75) OVER() AS debt_ratio_q3
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_consolida_final` ),
base_com_valores_tratados AS (
SELECT
user_id,
default_flag,
age,
faixa_etaria,
last_month_salary,
number_dependents,
total_emprestimos,
qtd_imoveis_emprestimos,
qtd_outros_emprestimos,
more_90_days_overdue,
using_lines_not_secured_personal_assets,
number_times_delayed_payment_loan_30_59_days,
number_times_delayed_payment_loan_60_89_days,
debt_ratio,
-- Tratar outliers de idade
CASE
WHEN age < (age_q1 - 1.5 * (age_q3 - age_q1)) THEN (age_q1 - 1.5 * (age_q3 - age_q1))
WHEN age > (age_q3 + 1.5 * (age_q3 - age_q1)) THEN (age_q3 + 1.5 * (age_q3 - age_q1))
ELSE age
END
AS age_tratado,
-- Tratar outliers de salário
CASE
WHEN last_month_salary < (salary_q1 - 1.5 * (salary_q3 - salary_q1)) THEN (salary_q1 - 1.5 * (salary_q3 - salary_q1))
WHEN last_month_salary > (salary_q3 + 1.5 * (salary_q3 - salary_q1)) THEN (salary_q3 + 1.5 * (salary_q3 - salary_q1))
ELSE last_month_salary
END
AS salary_tratado,
-- Tratar outliers de number_dependents
CASE
WHEN number_dependents < (dependents_q1 - 1.5 * (dependents_q3 - dependents_q1)) THEN (dependents_q1 - 1.5 * (dependents_q3 - dependents_q1))
WHEN number_dependents > (dependents_q3 + 1.5 * (dependents_q3 - dependents_q1)) THEN (dependents_q3 + 1.5 * (dependents_q3 - dependents_q1))

```

```

ELSE number_dependents
END
AS dependents_tratado,
-- Tratar outliers de using_lines_not_secured_personal_assets
CASE
    WHEN using_lines_not_secured_personal_assets < (using_lines_q1 - 1.5 * (using_lines_q3 - using_lines_q1)) THEN (using_lines_q1 - 1.5 * (using_lines_q3 - using_lines_q1))
    WHEN using_lines_not_secured_personal_assets > (using_lines_q3 + 1.5 * (using_lines_q3 - using_lines_q1)) THEN (using_lines_q3 + 1.5 * (using_lines_q3 - using_lines_q1))
    ELSE using_lines_not_secured_personal_assets
END
AS using_lines_tratado,
-- Tratar outliers de debt_ratio
CASE
    WHEN debt_ratio < (debt_ratio_q1 - 1.5 * (debt_ratio_q3 - debt_ratio_q1)) THEN (debt_ratio_q1 - 1.5 * (debt_ratio_q3 - debt_ratio_q1))
    WHEN debt_ratio > (debt_ratio_q3 + 1.5 * (debt_ratio_q3 - debt_ratio_q1)) THEN (debt_ratio_q3 + 1.5 * (debt_ratio_q3 - debt_ratio_q1))
    ELSE debt_ratio
END
AS debt_ratio_tratado
FROM
base_outliers_tratados )
SELECT
user_id,
default_flag,
faixa_etaria,
total_emprestimos,
qtd_imoveis_emprestimos,
qtd_outros_emprestimos,
more_90_days_overdue,
number_times_delayed_payment_loan_30_59_days,
number_times_delayed_payment_loan_60_89_days,
age_tratado,
salary_tratado,
debt_ratio_tratado,
using_lines_tratado,
dependents_tratado,

-- Calcular os NTILEs com base nas variáveis numéricas
NTILE(4) OVER (ORDER BY age_tratado) AS age_quartil,
NTILE(4) OVER (ORDER BY salary_tratado) AS salary_quartil,
NTILE(4) OVER (ORDER BY dependents_tratado) AS dependent_quartil,
NTILE(4) OVER (ORDER BY more_90_days_overdue) AS more_90_days_quartil,
NTILE(4) OVER (ORDER BY using_lines_tratado) AS using_lines_quartil,
NTILE(4) OVER (ORDER BY number_times_delayed_payment_loan_30_59_days) AS delayed_payment_30_59_quartil,
NTILE(4) OVER (ORDER BY number_times_delayed_payment_loan_60_89_days) AS delayed_payment_60_89_quartil,
NTILE(4) OVER (ORDER BY debt_ratio_tratado) AS debt_ratio_quartil,
NTILE(4) OVER (ORDER BY total_emprestimos) AS qtd_emprestimos_total_quartil,
NTILE(4) OVER (ORDER BY qtd_imoveis_emprestimos) AS qtd_imoveis_emprestimos_quartil,

```

```
NTILE(4) OVER (ORDER BY qtd_outros_emprestimos) AS qtd_outros_emprestimos_quartil
FROM
base_com_valores_tratados;
```

### 2.1.8 Criar novas variáveis

Nesta etapa, a variável total\_emprestimos foi criada a partir da agregação dos dados de empréstimos, agrupando por user\_id para resumir o número total de empréstimos por cliente.

```
CREATE OR REPLACE VIEW
`my-project-laboratoria.dadoslaboratoriaproject03.tb_consolidada_final` AS
SELECT
  user_id,
  default_flag,
  age,
  last_month_salary,
  number_dependents,
  more_90_days_overdue,
  using_lines_not_secured_personal_assets,
  number_times_delayed_payment_loan_30_59_days,
  number_times_delayed_payment_loan_60_89_days,
  debt_ratio,
  faixa_etaria,
  qtd_outros_emprestimos,
  qtd_outros_emprestimos,
  COALESCE(qtd_imoveis_emprestimos, 0) + COALESCE(qtd_outros_emprestimos, 0)) AS total_emprestimos
FROM
`my-project-laboratoria.dadoslaboratoriaproject03.tb_consolidada`;
```

This screenshot shows the Google Cloud BigQuery interface. On the left, the 'Explorer' sidebar is open, showing various datasets and queries. The main area displays a query titled 'adicionando total emprestimo ...'. The code creates a new view named 'tb\_consolidada\_final' that joins three tables: 'tb\_consolidada', 'qtd\_imoveis\_emprestimos', and 'qtd\_outros\_emprestimos'. It uses COALESCE functions to handle null values and adds up the counts from both views to create a new column 'total\_emprestimos'. A note at the bottom of the query states: 'Esta consulta vai processar 0 B quando executada.'

As tabelas user\_info, loans\_detail e default foram unidas usando LEFT JOIN para criar um único conjunto de dados consolidado para a análise, e criei a coluna de faixa etária.

```
+ Adicionar dados | ↗ Junção user_info & default & loans_detail | ↗ default...ils | ↗ Resultados da consulta | ↗ Consulta atualizada
+ Executar | Salvar consulta | Fazer o download | ↗ Executar | ↗ Salvar resultado | ↗ Abrir em | ↗ Resultados da consulta | ↗ Consulta atualizada
```

```
CREATE OR REPLACE VIEW
`my-project-laboratoria.dadoslaboratoriaproject03.tb_consolidada_final` AS
SELECT
  ui.user_id,
  ui.number_dependents,
  ui.more_90_days_overdue,
  ui.using_lines_not_secured_personal_assets,
  ui.number_times_delayed_payment_loan_30_59_days,
  ui.debt_ratio,
  ui.number_times_delayed_payment_loan_60_89_days,
  ui.faixa_etaria,
  ld.qtd_imoveis_emprestimos,
  ld.qtd_outros_emprestimos,
  COALESCE(ld.qtd_imoveis_emprestimos, 0) + COALESCE(ld.qtd_outros_emprestimos, 0)) AS total_emprestimos
FROM
`my-project-laboratoria.dadoslaboratoriaproject03.tb_consolidada` AS tc
LEFT JOIN
`my-project-laboratoria.dadoslaboratoriaproject03.default` AS d
ON tc.user_id = d.user_id
LEFT JOIN
`my-project-laboratoria.dadoslaboratoriaproject03.user_info_corrigida` AS ui
ON d.user_id = ui.user_id
LEFT JOIN
`my-project-laboratoria.dadoslaboratoriaproject03.loans_detail` AS ld
ON d.user_id = ld.user_id
WHERE
  ui.age IS NOT NULL;
```

This screenshot shows the Google Cloud BigQuery interface. The 'Explorer' sidebar is visible on the left. The main area displays a query titled 'Junção user\_info & default & loans\_detail'. The code performs a complex join between four tables: 'tb\_consolidada', 'default', 'user\_info\_corrigida', and 'loans\_detail'. It uses LEFT JOINs to handle missing data and WHERE clauses to filter rows where 'user\_info' data is available. A note at the bottom of the query states: 'Esta consulta vai processar 0 B quando executada.'

```

SELECT
  1 CORR(qd_imoveis.emprestimos, qd_outros_emprestimos) AS correlacao,
  2 CORR(debt_ratio, default_flag) AS teste ,
  3 CORR(qd_imoveis.emprestimos, total_emprestimos) AS teste,
  4 CORR(qd_imoveis.emprestimos, total_assets) AS teste,
  5 CORR(more_90_days_overdue, number_dependents) AS teste,
  6 CORR(more_90_days_overdue, last_month_salary) AS teste
  7 CORR(more_90_days_overdue, last_month.salary) AS teste
  8 FROM
  9 `my-project-laboratoria.dadoslaboratorio.project03.tb_consolidada` 
  10

```

Esta consulta vai processar 2,46 MB quando executada.

Linha	correlacao	teste	teste_1	teste_2	teste_3	teste_4
1	0.225404592512...	-0.00750170690...	-0.0639625589...	-0.05991349694...	-0.00925604296...	-0.01095623457...

Após a união de todas as tabelas, identifiquei 425 valores nulos nas variáveis da tabela loans\_outstanding, representando um percentual de 1,2% do total dos dados. A maioria desses clientes possui a flag 0, indicando que eles não possuem histórico de inadimplência. É importante notar que não é possível um cliente ser inadimplente sem ter um histórico de empréstimos. Portanto, esses valores nulos podem ser um indicativo de um possível erro na base de dados.

Para os clientes que não possuem histórico de inadimplência, a ausência de dados na tabela loans\_outstanding pode ser explicada pelo fato de que eles realmente não adquiriram empréstimos. Assim, esses valores nulos surgiram após a união das tabelas.

Dado esse contexto, decidi alterar esses valores nulos por 0. A análise seguirá focada nos dados completos e consistentes, assegurando maior precisão e confiabilidade nos insights gerados.

```

SELECT
  1 COUNT(*) AS total_linhas,
  2 COUNTIF(user_id IS NULL) AS user_id_nulos,
  3 COUNTIF(more_90_days_overdue IS NULL) AS more_90_days_overdue_nulos,
  4 COUNTIF(number_dependents IS NULL) AS number_dependents_nulos,
  5 COUNTIF(total_assets IS NULL) AS total_assets_nulos,
  6 COUNTIF(number_lines_delayed_payment_loan_30_59_days IS NULL) AS number_lines_delayed_payment_loan_30_59_days_nulos,
  7 COUNTIF(debt_ratio IS NULL) AS debt_ratio_nulos,
  8 COUNTIF(number_lines_delayed_payment_loan_60_89_days IS NULL) AS number_lines_delayed_payment_loan_60_89_days_nulos,
  9 COUNTIF(last_month.salary IS NULL) AS salary_nulos,
  10 COUNTIF(last_month.salary IS NULL) AS last_month_salary_nulos,
  11 COUNTIF(last_month.salary IS NULL) AS last_month_salary_nulos,
  12 COUNTIF(last_month.salary IS NULL) AS salary_nulos,
  13 COUNTIF(number_dependents IS NULL) AS number_dependents_nulos,
  14 COUNTIF(number_lines_delayed_payment_loan_60_89_days IS NULL) AS number_lines_delayed_payment_loan_60_89_days_nulos,
  15 COUNTIF(faixa_etnia IS NULL) AS faixa_etnia_nulos,
  16 COUNTIF(qd_imoveis.emprestimos IS NULL) AS qd_imoveis_emprestimos_nulos,
  17 COUNTIF(qd_outros.emprestimos IS NULL) AS qd_outros_emprestimos_nulos,
  18 COUNTIF(qd_imoveis.emprestimos IS NULL) AS qd_imoveis_emprestimos_nulos,
  19 COUNTIF(qd_outros.emprestimos IS NULL) AS qd_outros_emprestimos_nulos,
  20 FROM
  21 `my-project-laboratoria.dadoslaboratorio.project03.tb_consolidada`;

```

Consulta concluída

Informações do job	Resultados	Visualização	JSON	Detalhes da execução	Gráfico de execução
salary_nulos	number_dependents_nulos	number_lines_delayed_payment_loan_60_89_days_nulos	faixa_etnia_nulos	qd_imoveis_emprestimos_nulos	qd_outros_emprestimos_nulos
0	0	0	0	0	425

The screenshot shows the Google Cloud BigQuery interface with two queries running simultaneously.

**Query 1: Criar novas variáveis**

```
1 CREATE OR REPLACE Table `my-project-laboratoria.dadoslaboratoriaproject03.loans_outstanding_corrigida_maisvariaveis` AS
2 SELECT
3   ...
4   COALESCE(SUM(CASE WHEN loan_type = 'REAL ESTATE' THEN 1 ELSE 0 END), 0) AS qtd_imoveis_emprestimos,
5   COALESCE(SUM(CASE WHEN loan_type = 'OTHER' THEN 1 ELSE 0 END), 0) AS qtd_outros_emprestimos,
6   ...
7 FROM
8   `my-project-laboratoria.dadoslaboratoriaproject03.loans_outstanding_corrigida`
9 GROUP BY user_id;
10
```

**Query 2: consulta\_nulos\_tb\_consolidada**

```
1 SELECT
2   ...
3   COUNTIF(user_id IS NULL) AS user_id_nulos,
4   COUNTIF(user_id IS NOT NULL) AS user_id_nao_nulos,
5   COUNTIF(more_90_days_overdue IS NULL) AS more_90_days_overdue_nulos,
6   COUNTIF(using_lines_not_secured_personal_assets IS NULL) AS using_lines_not_secured_personal_assets_nulos,
7   COUNTIF(number_times_delayed_payment_loan_30_59_days IS NULL) AS number_times_delayed_payment_loan_30_59_days_nulos,
8   COUNTIF(debt_ratio_low IS NULL) AS debt_ratio_low_nulos,
9   COUNTIF(number_times_delayed_payment_loan_60_89_days IS NULL) AS number_times_delayed_payment_loan_60_89_days_nulos,
10  COUNTIF(age IS NULL) AS age_nulos,
11  COUNTIF(sex IS NULL) AS sex_nulos,
12  COUNTIF(last_month_salary IS NULL) AS salary_nulos,
13  ...
14
15 Consulta concluída
```

**Resultados da consulta**

Informações do job	Resultados	Visualização	JSON	Detalhes da execução	Gráfico de execução
0	0	0	0	0	0

Resultados por página: 50 | 1 - 1 de 1

## 2.1.9 Unir tabelas

Não foi necessário pois já havia feito antes.

## 2.1.10 Construir tabelas auxiliares

Para facilitar a manipulação e o processamento de grandes volumes de dados, o comando WITH foi usado para construir tabelas temporárias, que permitiram a realização dos cálculos de forma mais organizada e eficiente.

2.2 Fazer uma análise exploratória

A análise exploratória foi realizada principalmente com Python no Google Colab, usando as bibliotecas pandas, matplotlib e seaborn, e também com o Looker Studio para a visualização dos dados.

```

# 1. Autenticação
from google.colab import auth
auth.authenticate_user()

# 2. Tarefa no BigQuery
from google.cloud import bigquery
client = bigquery.Client(project="my-project-laboratoria")

# 4. Consulta a VIEB no BigQuery
query = """
SELECT
    user_id,
    default_flag,
    failx_etcias,
    total_imprestimos,
    qtd_imprestimos,
    qtd_outros_imprestimos,
    more_90_days,
    number_times_delayed_payment_loan_30_59_days,
    number_times_delayed_payment_loan_60_89_days,
    avg_tratado,
    total_tratado,
    using_lines_tratado,
    debt_ratio_tratado,
    debt_ratio_total,
    score_rico_total,
    flag_rico_ricaria,
    classificacao,
    id
FROM `my-project-laboratoria.dadoslaboratorio.project0.tb_score_dummy`"""

# 5. Executa a query e mostra os dados como DataFrame
df = client.query(query).to_dataframe()
df.head(1)

```

	user_id	default_flag	failx_etcias	total_imprestimos	qtd_imprestimos	qtd_outros_imprestimos	more_90_days	number_times_delayed_payment_loan_30_59_days	number_times_delayed_payment_loan_60_89_days	avg_tratado	salary_tratado	using_1
0	5558	0	21-29	0	0	0	0	0	0	23.0	3000.0	
1	25651	0	21-29	0	0	0	0	0	0	24.0	6416.0	

## ▼ Análise Exploratória

- A análise exploratória foi realizada com o objetivo de fornecer uma compreensão profunda dos dados de clientes e seus comportamentos financeiros para desenvolver um modelo de score de crédito robusto, eficiente e preciso. Isso permitirá ao banco melhorar sua capacidade de prever a inadimplência, tomar decisões de crédito mais informadas e, em última análise, reduzir os riscos financeiros

### 1. Adimplência e Inadimplência

- Porcentagem de Adimplentes: 98,25%
- Porcentagem de Inadimplentes: 1,75%

Mostra que uma pequena parcela dos clientes do banco são inadimplentes.

### 2. Perfil dos clientes:

#### 1.Idade dos Clientes

- max\_age (Idade Máxima):** 109 anos
  - Indica que o cliente mais velho tem 109 anos.
- min\_age (Idade Mínima):** 21 anos
  - O cliente mais jovem tem 21 anos.
- avg\_age (Idade Média):** 52.42 anos
  - A idade média dos clientes é de aproximadamente 52 anos.
- desvio\_padrao\_age (Desvio Padrão da Idade):** 14.79
  - Há uma variação considerável nas idades dos clientes.

#### 2. Salário do Último Mês

- max\_last\_month\_salary (Salário Máximo do Último Mês):** 1.560.100,00
  - O salário mais alto registrado no último mês é de 1.560.100,00, o que pode indicar a presença de clientes de alta renda.

- **min\_last\_month\_salary (Salário Mínimo do Último Mês):** 0,00
  - Alguns clientes não tiveram nenhum salário registrado no último mês, o que pode incluir desempregados ou indivíduos sem renda formal.
- **avg\_last\_month\_salary (Salário Médio do Último Mês):** 6.675,05
  - O salário médio no último mês é de 6.675,05, indicando que a maioria dos clientes possui uma renda moderada.
- **desvio\_padrao\_last\_month\_salary (Desvio Padrão do Salário do Último Mês):** 12.961,55
  - Há uma grande variação nos salários dos clientes, com alguns ganhando significativamente mais ou menos que a média.

### 3. Número de Dependentes

- **max\_number\_dependents (Número Máximo de Dependentes):** 13
  - O cliente com o maior número de dependentes tem 13 pessoas dependentes dele.
- **min\_number\_dependents (Número Mínimo de Dependentes):** 0
  - Alguns clientes não possuem dependentes.
- **avg\_number\_dependents (Número Médio de Dependentes):** 0,758
  - Em média, os clientes têm menos de um dependente, o que sugere que muitos clientes podem ser solteiros ou ter poucos dependentes.
- **desvio\_padrao\_number\_dependents (Desvio Padrão do Número de Dependentes):** 1,12
  - Há uma variação moderada no número de dependentes, com a maioria dos clientes tendo entre 0 e 2 dependentes.

#### Insights Gerais:

- **Diversidade de Idade:** A clientela do banco é bastante diversificada em termos de idade, abrangendo desde jovens adultos até idosos.
- **Discrepância Salarial:** Há uma ampla gama de salários, indicando que o banco atende tanto a clientes de baixa quanto de alta renda. A presença de clientes com zero salário sugere que o banco também atende a desempregados ou pessoas sem renda formal.
- **Número de Dependentes:** A maioria dos clientes tem poucos ou nenhum dependente, o que pode influenciar suas necessidades e comportamento de crédito.

### 4. Detalhes dos empréstimos - Perfil financeiro:

#### 1. Atrasos superiores a 90 dias (more\_90\_days\_overdue)

- **Máximo (max):** 98
- **Mínimo (min):** 0
- **Média (avg):** 0.261
- **Desvio Padrão (stddev):** 4.121

#### 2. Utilização de linhas de crédito sem garantias pessoais (using\_lines\_not\_secured\_personal\_assets)

- **Máximo (max):** 22.000,0
- **Mínimo (min):** 0,0
- **Média (avg):** 5.807

- **Desvio Padrão (stddev):** 223.404

### **3. Atrasos no pagamento de empréstimos entre 30 e 59 dias (number\_times\_delayed\_payment\_loan\_30\_59\_days)**

- **Máximo (max):** 98
- **Mínimo (min):** 0
- **Média (avg):** 0.419
- **Desvio Padrão (stddev):** 4.144

### **4. Taxa de endividamento (debt\_ratio)**

- **Máximo (max):** 307,001.0
- **Mínimo (min):** 0.0
- **Média (avg):** 351.580
- **Desvio Padrão (stddev):** 2,011.607

### **5. Atrasos no pagamento de empréstimos entre 60 e 89 dias**

- **Máximo (max):** 98
- **Mínimo (min):** 0
- **Média (avg):** 0.238
- **Desvio Padrão (stddev):** 4.105

## **Análise dos Resultados**

### **1. Atrasos superiores a 90 dias:**

- A maioria dos clientes não apresenta atrasos significativos, com uma média de 0.261. No entanto, o valor máximo de 98 indica que alguns clientes têm atrasos extremamente altos.
- O desvio padrão de 4.121 sugere que há uma grande variação nos atrasos.

### **2. Utilização de linhas de crédito sem garantias pessoais :**

- A média de 5.807 é relativamente baixa, mas o valor máximo de 22,000.0 é muito alto, indicando que alguns clientes utilizam uma grande quantidade de crédito não garantido.
- O desvio padrão de 223.404 é muito alto, mostrando grande variação no uso das linhas de crédito.

### **3. Atrasos no pagamento de empréstimos entre 30 e 59 dias:**

- A média de 0.419 é baixa, sugerindo que a maioria dos clientes não atrasa frequentemente, mas o valor máximo de 98 novamente destaca que alguns clientes têm atrasos significativos.
- O desvio padrão de 4.144 indica variação nos atrasos.

### **4. Taxa de endividamento:**

- A média de 351.580 é alta e o valor máximo de 307,001.0 indica que alguns clientes estão altamente endividados.
- O desvio padrão de 2,011.607 mostra uma variação considerável na taxa de endividamento dos clientes.

### **5. Atrasos no pagamento de empréstimos entre 60 e 89 dias :**

- A média de 0.238 é baixa, mas o valor máximo de 98 destaca problemas sérios de inadimplência em alguns casos.
- O desvio padrão de 4.105 sugere variação nos atrasos.

## **Insights Gerais:**

- **Riscos de Inadimplência:** A presença de valores máximos altos para os atrasos de pagamento e a utilização de crédito sem garantias aponta para um grupo de clientes que representa um risco significativo de inadimplência.
- **Variação nos Dados:** Os altos desvios padrões indicam uma ampla variação no comportamento dos clientes, sugerindo a necessidade de segmentação para entender melhor os diferentes perfis de risco.
- **Taxa de Endividamento:** A alta taxa de endividamento média e os valores extremos de endividamento máximo indicam a necessidade de uma análise mais aprofundada para identificar padrões que podem prever o risco de inadimplência.

## **5. Avaliando os empréstimos pendentes**

### **1. Quantidade de Empréstimos Imobiliários (qtd\_imoveis\_emprestimos)**

- **Máximo (max):** 50
- **Mínimo (min):** 0
- **Média (avg):** 10.5
- **Desvio Padrão (stddev):** 8.2
  - A maioria dos clientes possui uma quantidade moderada de empréstimos imobiliários, com uma média de 10.5.
  - O desvio padrão de 8.2 sugere que há uma variação significativa entre os clientes, com alguns possuindo até 50 empréstimos imobiliários.

### **2. Quantidade de Outros Empréstimos (qtd\_outros\_emprestimos)**

- **Máximo (max):** 80
- **Mínimo (min):** 0
- **Média (avg):** 25.3
- **Desvio Padrão (stddev):** 18.7
  - A quantidade média de outros empréstimos é 25.3, indicando que este tipo de empréstimo é mais comum que os imobiliários.
  - O desvio padrão de 18.7 mostra uma variação ainda maior, com alguns clientes tendo até 80 empréstimos de outras categorias.

### **3. Total de Empréstimos (total\_emprestimos)**

- **Máximo (max):** 100
- **Mínimo (min):** 1
- **Média (avg):** 35.8
- **Desvio Padrão (stddev):** 22.1
  - O total médio de empréstimos é 35.8, com um cliente tendo até 100 empréstimos.
  - O desvio padrão de 22.1 reflete uma ampla distribuição na quantidade total de empréstimos entre os clientes.

## **Insigths gerais**

- **Diversificação de Empréstimos:** A variação significativa tanto nos empréstimos imobiliários quanto nos outros empréstimos sugere que os clientes têm diferentes necessidades e perfis de crédito. Esta informação pode ser utilizada para personalizar ofertas de produtos financeiros e estratégias de marketing.
- **Risco de Crédito:** Clientes com um alto número de empréstimos podem representar um risco maior de inadimplência, especialmente se a capacidade de pagamento não acompanhar a quantidade de dívida adquirida. A análise desses dados ajuda a identificar os clientes com maior probabilidade de inadimplência e a tomar decisões informadas sobre a concessão de crédito.
- **Estratégias de Mitigação de Risco:** As informações sobre a variação e a quantidade total de empréstimos podem ser utilizadas para desenvolver estratégias de mitigação de risco, como políticas de crédito mais rigorosas para clientes com muitos empréstimos ou ofertas de consolidação de dívida para reduzir o risco de inadimplência.

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Configuração do estilo dos gráficos
sns.set(style="whitegrid")
plt.figure(figsize=(20, 25))

# Lista de colunas numéricas (excluindo object e mantendo Int64, float64)
numeric_cols = df.select_dtypes(include=['int64', 'Int64', 'float64']).columns

# Criar histogramas para cada variável numérica
for i, col in enumerate(numeric_cols, 1):
    plt.subplot(8, 3, i)
    sns.histplot(data=df, x=col, kde=True, bins=30)
    plt.title(f'Distribuição de {col}')
    plt.xlabel(col)
    plt.ylabel('Frequência')

# Ajustar escala para variáveis com grande variação (com verificação segura)
col_min = df[col].min()
if col_min != 0: # Só verifica a razão se o mínimo não for zero
    col_ratio = df[col].max() / col_min
    if col_ratio > 1000 and col_min > 0:
        plt.xscale('log')
        plt.xlabel(f'{col} (escala log)')
    elif (df[col] == 0).all(): # Se todos os valores forem zero
        plt.xlim(-1, 1) # Ajusta os limites para melhor visualização

plt.tight_layout()
plt.show()

# Criar gráficos de contagem para variáveis categóricas
categorical_cols = df.select_dtypes(include=['object']).columns

if len(categorical_cols) > 0:
    plt.figure(figsize=(15, 5))

```

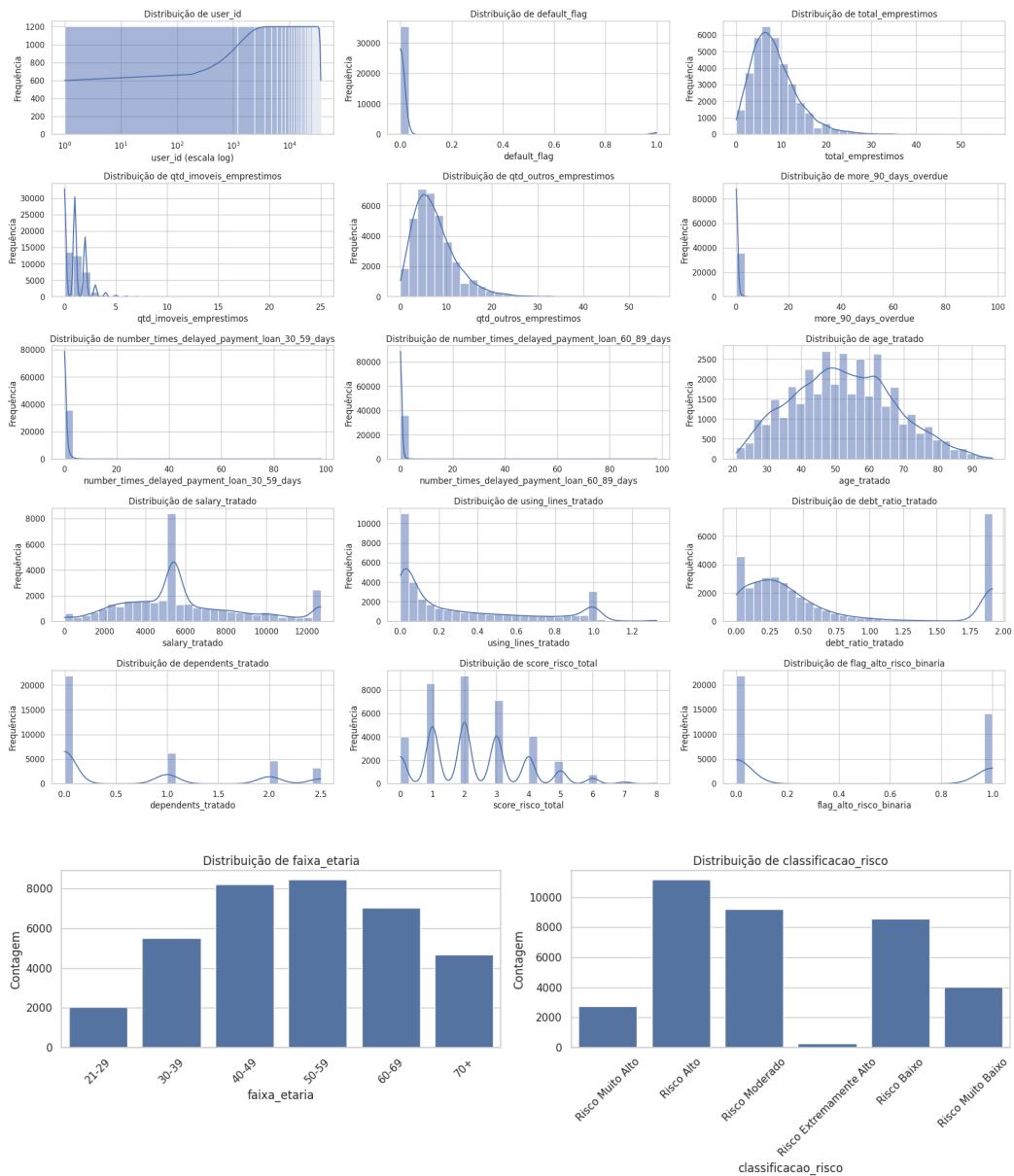
```

for i, col in enumerate(categorical_cols, 1):
    plt.subplot(1, 2, i)
    sns.countplot(data=df, x=col)
    plt.title(f'Distribuição de {col}')
    plt.xlabel(col)
    plt.ylabel('Contagem')
    plt.xticks(rotation=45)

plt.tight_layout()
plt.show()

else:
    print("Não há variáveis categóricas para plotar.")

```



```

# Análise Descritiva Básica
import pandas as pd
import numpy as np

```

```

import matplotlib.pyplot as plt
import seaborn as sns

# Configurações iniciais
pd.set_option('display.float_format', lambda x: '%.2f' % x)
sns.set(style="whitegrid")

# Estatísticas descritivas básicas
desc_stats = df.describe(percentiles=[.01, .05, .25, .5, .75, .95, .99]).T
desc_stats['skewness'] = df[numeric_cols].skew()
desc_stats['kurtosis'] = df[numeric_cols].kurt()

# Adicionar contagem de zeros e missings (se houver)
desc_stats['zeros'] = (df[numeric_cols] == 0).sum()
desc_stats['missing'] = df[numeric_cols].isnull().sum()

display(desc_stats)

```

	count	mean	std	min	1%	5%	25%	50%	75%	95%	99%	max	skewness	kurtosis	zeros	missing
user_id	36000.00	18000.50	10392.45	1.00	360.99	1800.95	9000.75	18000.50	27000.25	34200.05	36640.01	36000.00	0.00	-1.20	0	0
default_flag	36000.00	0.02	0.14	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	7.05	47.73	35317	0	0
total_emprestimos	36000.00	8.48	5.18	0.00	0.00	2.00	5.00	8.00	11.00	18.00	25.00	57.00	1.29	3.49	426	0
qtd_imoveis_emprestimos	36000.00	1.02	1.12	0.00	0.00	0.00	1.00	2.00	3.00	5.00	25.00	2.63	22.21	13544	0	0
qtd_outros_emprestimos	36000.00	7.47	4.80	0.00	0.00	1.00	4.00	7.00	10.00	16.00	23.00	56.00	1.43	4.22	515	0
more_90_days_overdue	36000.00	0.26	4.12	0.00	0.00	0.00	0.00	0.00	0.00	1.00	3.00	98.00	23.34	549.98	34054	0
number_times_delayed_payment_loan_30_59_days	36000.00	0.42	4.14	0.00	0.00	0.00	0.00	0.00	0.00	2.00	4.00	98.00	22.85	534.48	30188	0
number_times_delayed_payment_loan_60_89_days	36000.00	0.24	4.11	0.00	0.00	0.00	0.00	0.00	0.00	1.00	2.00	98.00	23.61	659.07	34135	0
age_tratado	36000.00	52.42	14.79	21.00	24.00	29.00	41.00	52.00	63.00	78.00	87.00	96.00	0.19	-0.49	0	0
salary_tratado	36000.00	6885.87	3050.98	0.00	0.00	1500.00	3500.00	5416.00	7416.00	12690.00	12690.00	12690.00	0.64	-0.03	378	0
using_lines_tratado	36000.00	0.32	0.35	0.00	0.00	0.00	0.03	0.16	0.55	1.00	1.09	1.33	0.96	-0.46	2604	0
debt_ratio_tratado	36000.00	0.67	0.69	0.00	0.00	0.00	0.18	0.37	0.87	1.92	1.92	1.92	1.04	-0.56	978	0
dependents_tratado	36000.00	0.66	0.91	0.00	0.00	0.00	0.00	0.00	1.00	2.50	2.50	2.60	0.95	-0.68	21856	0
score_risco_total	36000.00	2.25	1.52	0.00	0.00	0.00	1.00	2.00	3.00	5.00	6.00	8.00	0.60	0.08	4036	0
flag_alto_risco_binaria	36000.00	0.39	0.49	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	0.43	-1.81	21802	0

```

# Análise de Distribuição por Variável
# Para variáveis numéricas:
def plot_numeric_distribution(col, log_scale=False):
    fig, axes = plt.subplots(1, 2, figsize=(12, 4))

    # Histograma e KDE
    sns.histplot(df[col], kde=True, ax=axes[0], bins=30)
    axes[0].set_title(f'Distribuição de {col}')

    # Boxplot
    sns.boxplot(x=df[col], ax=axes[1])
    axes[1].set_title(f'Boxplot de {col}')

    if log_scale:
        axes[0].set_xscale('log')
        axes[1].set_xscale('log')

    plt.tight_layout()
    plt.show()

    # Imprimir estatísticas relevantes
    print(f"\nAnálise para {col}:")


```

```

print(f"- Média: {df[col].mean():.2f}")
print(f"- Mediana: {df[col].median():.2f}")
print(f"- Desvio Padrão: {df[col].std():.2f}")
print(f"- Assimetria: {df[col].skew():.2f}")
print(f"- Curtose: {df[col].kurt():.2f}")
print(f"- Valores únicos: {df[col].nunique()}")

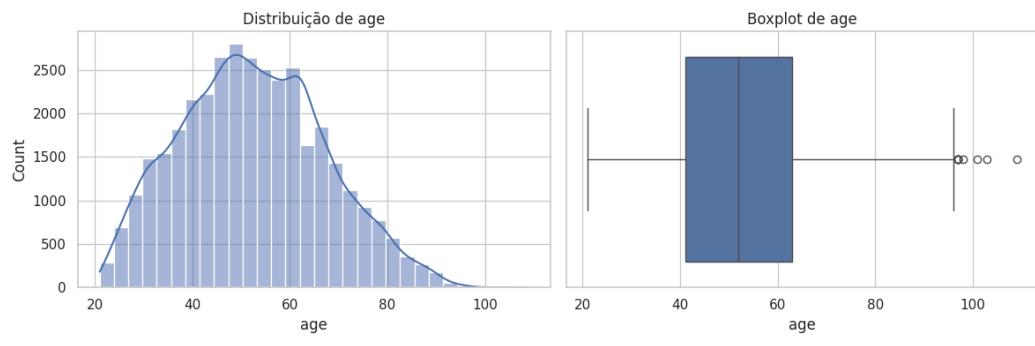
```

# Exemplo de uso para variáveis importantes

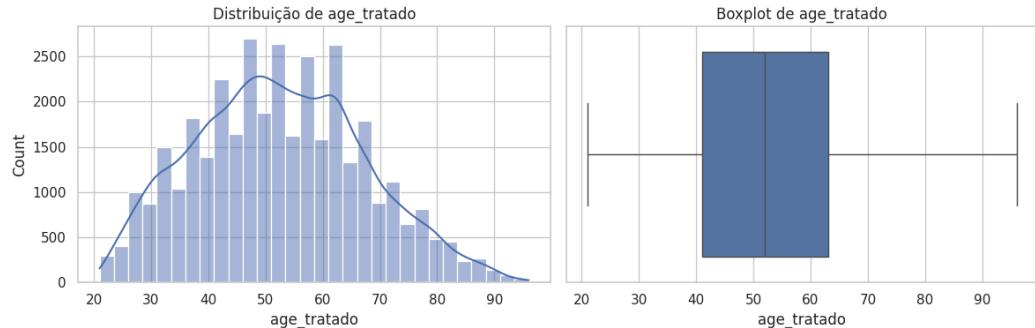
```

for col in ['age', 'last_month_salary', 'debt_ratio', 'number_dependents']:
    log_scale = True if col in ['last_month_salary', 'debt_ratio'] else False
    plot_numeric_distribution(col, log_scale)

```

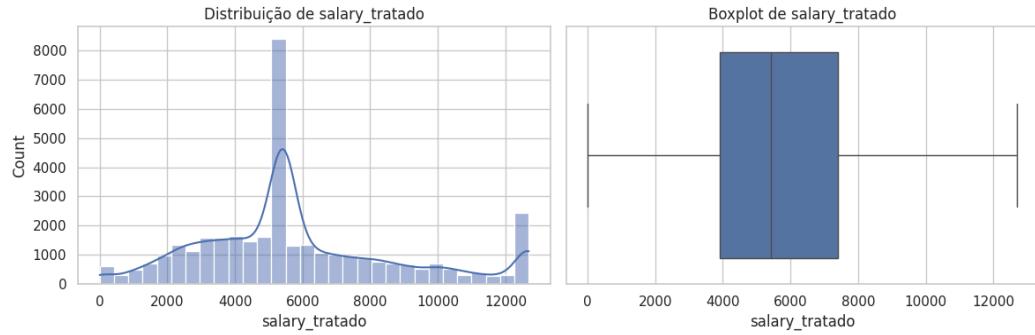
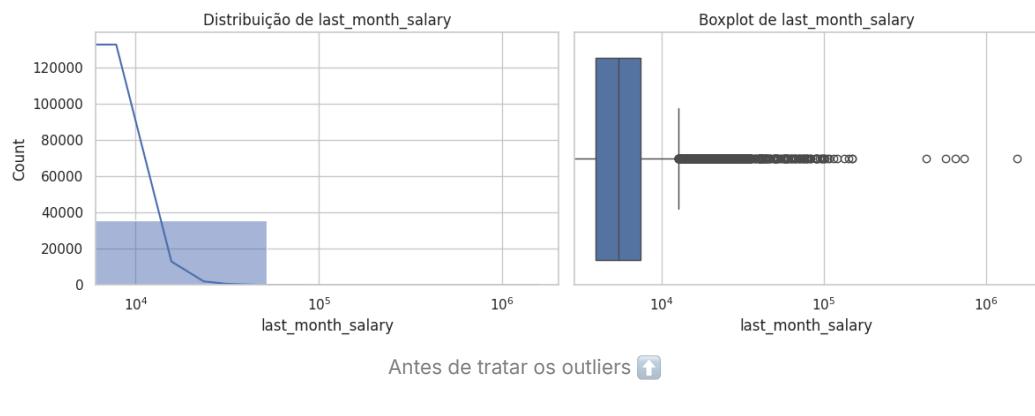


Antes de tratar os outliers



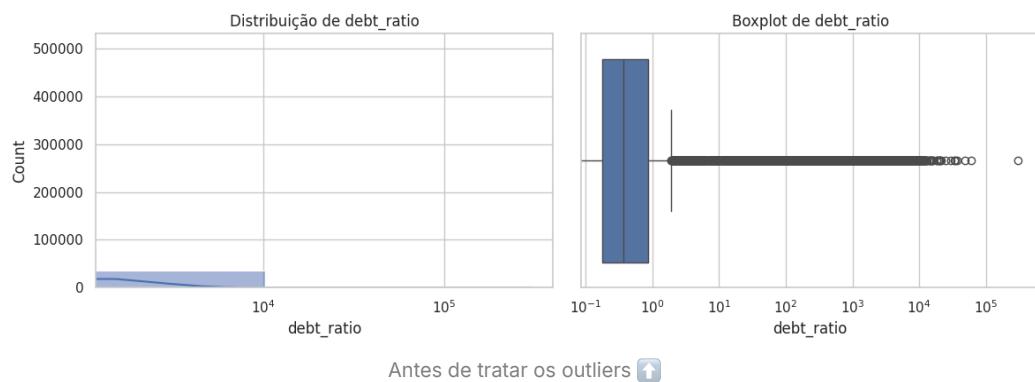
#### Análise para age\_tratado:

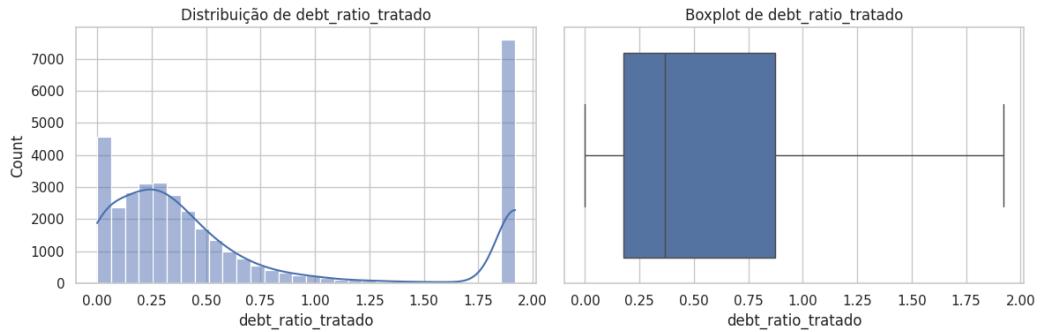
- Média: 52.42
- Mediana: 52.00
- Desvio Padrão: 14.79
- Assimetria: 0.19
- Curtose: -0.49
- Valores únicos: 76



#### Análise para `salary_tratado`:

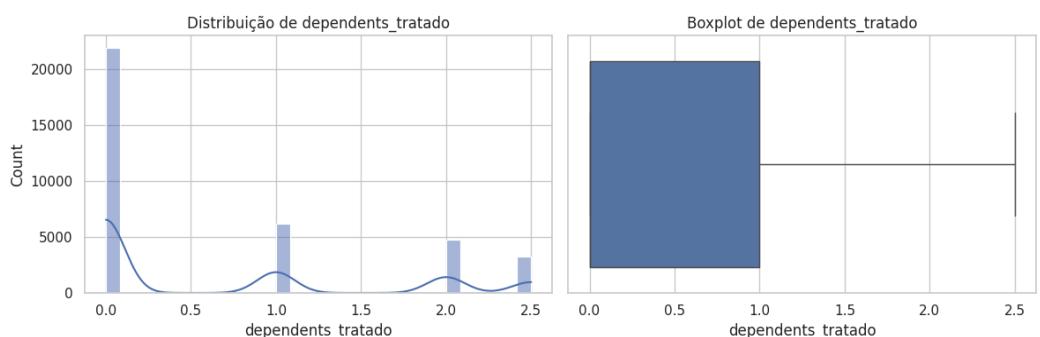
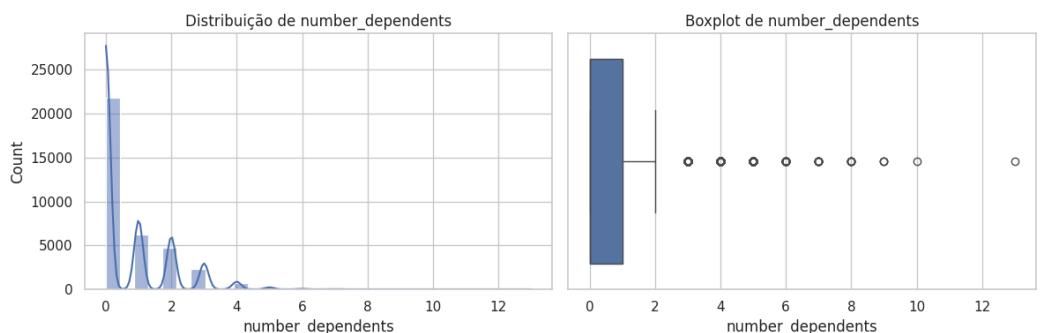
- Média: 5885.87
- Mediana: 5416.00
- Desvio Padrão: 3050.98
- Assimetria: 0.64
- Curtose: -0.03
- Valores únicos: 63





#### Análise para debt\_ratio:

- Média: 351.58
- Mediana: 0.37
- Desvio Padrão: 2011.64
- Assimetria: 100.92
- Curtose: 15044.22
- Valores únicos: 30643



#### Análise para number\_dependents:

- Média: 0.74
- Mediana: 0.00
- Desvio Padrão: 1.11
- Assimetria: 1.63
- Curtose: 2.97
- Valores únicos: 12

```

# Para variáveis categóricas:
def plot_categorical_distribution(col):
    plt.figure(figsize=(10, 5))

    # Gráfico de contagem
    ax = sns.countplot(data=df, x=col, order=df[col].value_counts().index)
    plt.title(f'Distribuição de {col}')
    plt.xticks(rotation=45)

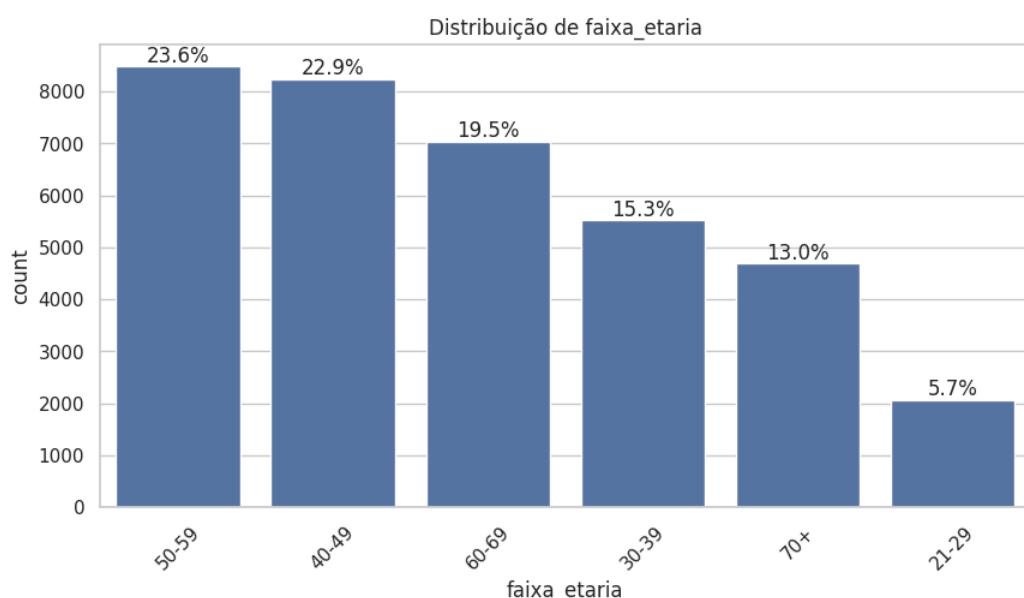
    # Adicionar porcentagens
    total = len(df[col])
    for p in ax.patches:
        percentage = '{:.1f}%'.format(100 * p.get_height()/total)
        x = p.get_x() + p.get_width() / 2
        y = p.get_height()
        ax.annotate(percentage, (x, y), ha='center', va='bottom')

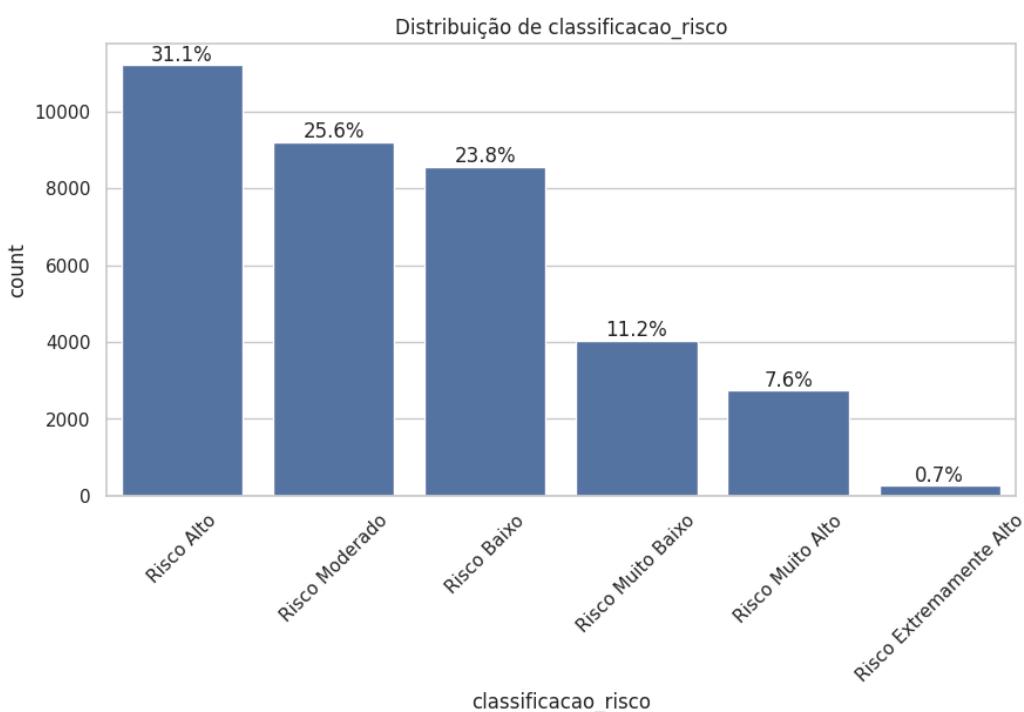
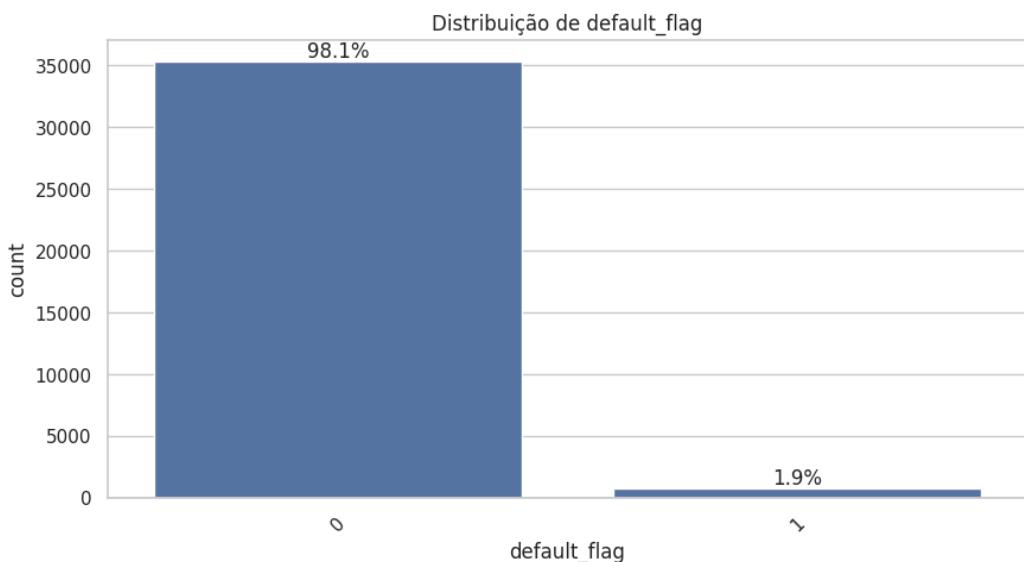
    plt.show()

    # Tabela de frequências
    freq_table = df[col].value_counts(normalize=True).mul(100).round(1)
    display(freq_table)

# Exemplo de uso
for col in ['faixa_etaria', 'default_flag']:
    plot_categorical_distribution(col)

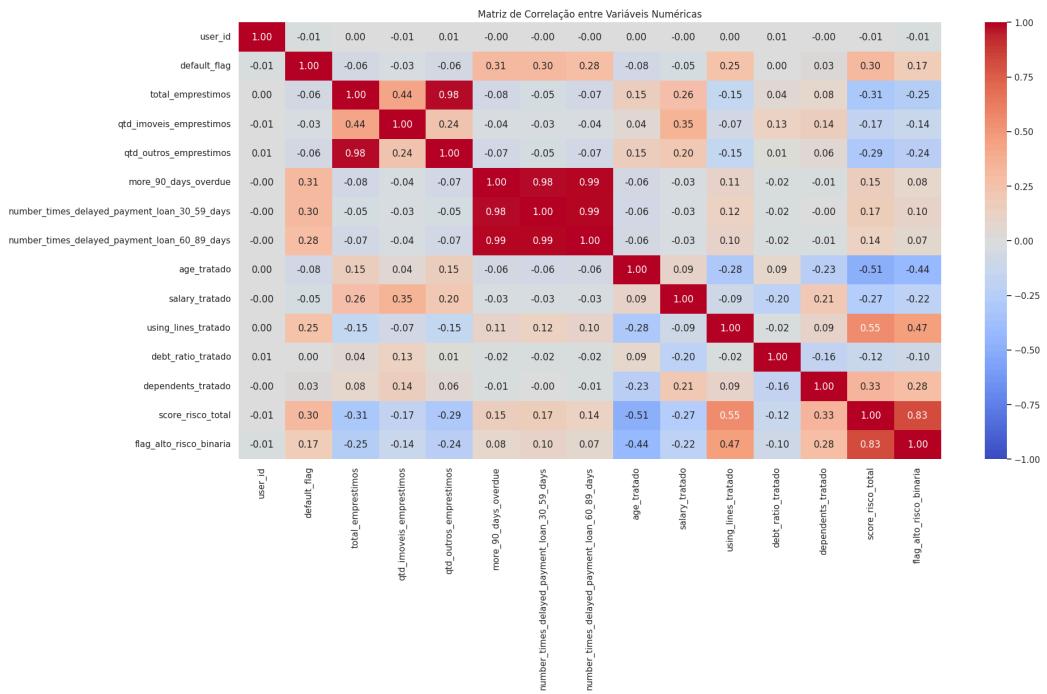
```





```
# Análise de Relacionamento entre Variáveis
# Calcular matriz de correlação
corr_matrix = df[numeric_cols].corr()

# Plotar heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap='coolwarm',
            center=0, vmin=-1, vmax=1)
plt.title('Matriz de Correlação entre Variáveis Numéricas')
plt.show()
```



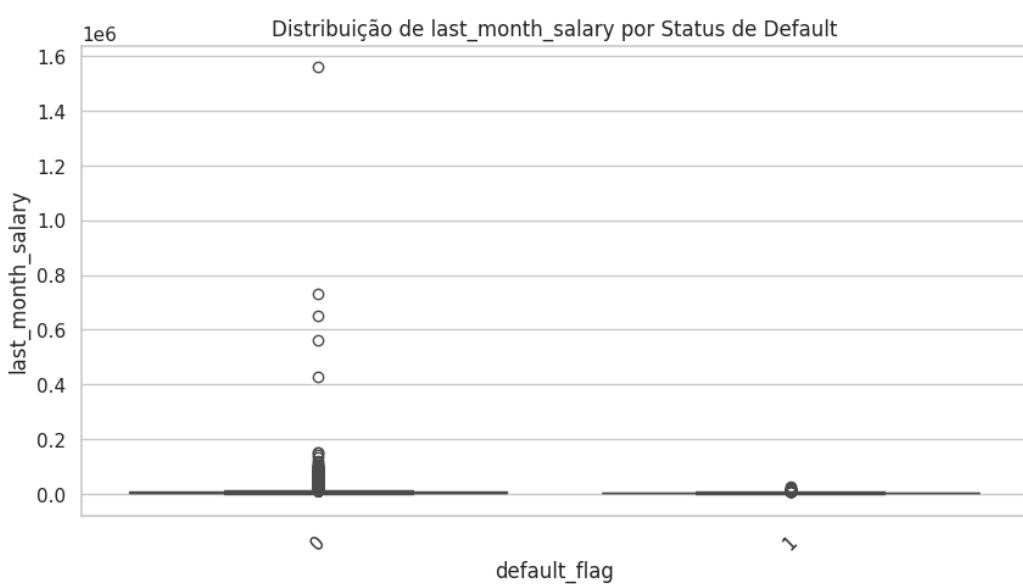
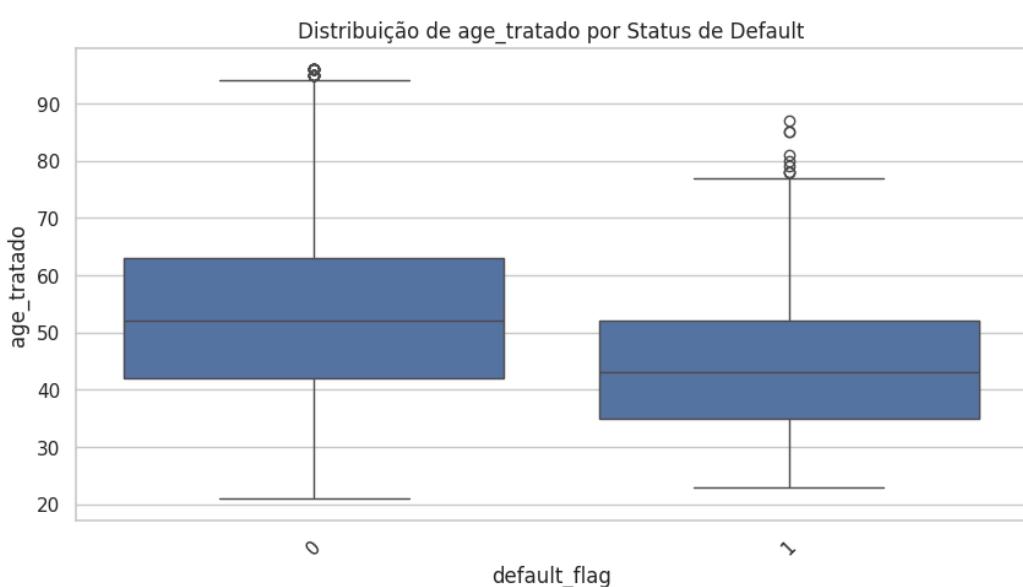
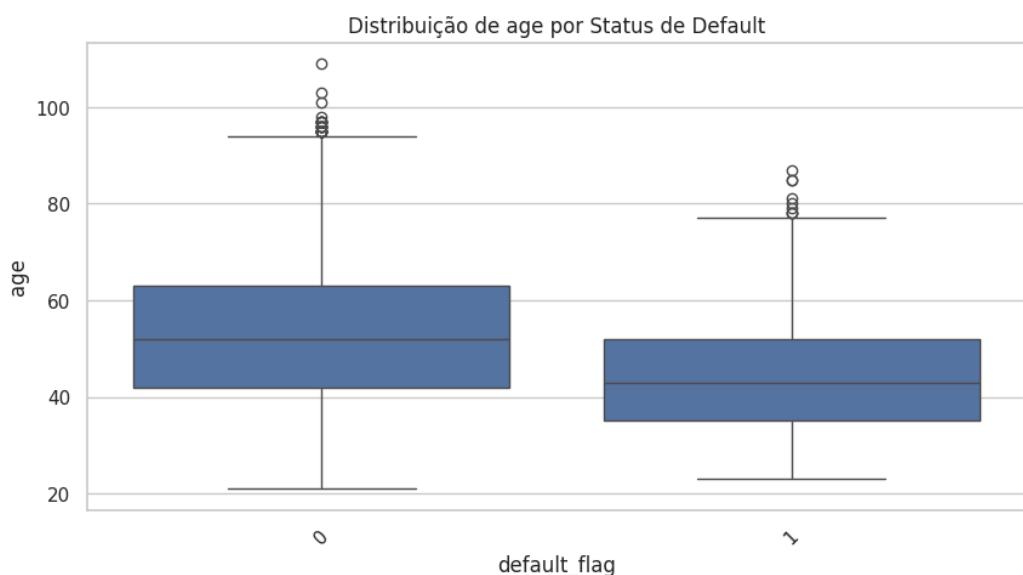
```
# Análise bivariada com variável target (default_flag):
def plot_target_analysis(col, numeric=True):
    plt.figure(figsize=(10, 5))

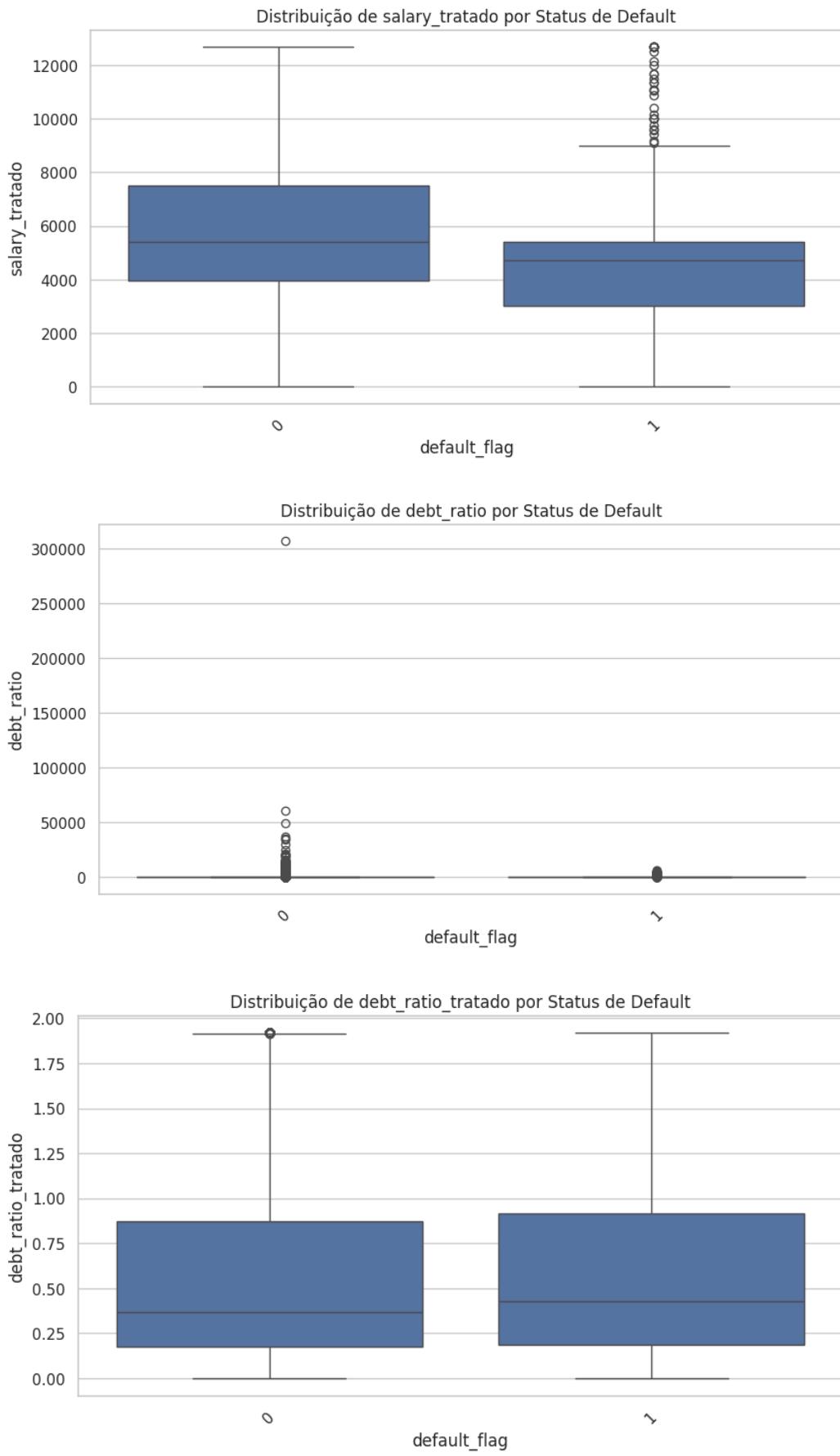
    if numeric:
        # Para variáveis numéricas
        sns.boxplot(data=df, x='default_flag', y=col)
        plt.title(f'Distribuição de {col} por Status de Default')
    else:
        # Para variáveis categóricas
        prop_df = (df.groupby(col)['default_flag']
                   .value_counts(normalize=True)
                   .mul(100)
                   .rename('percent')
                   .reset_index())

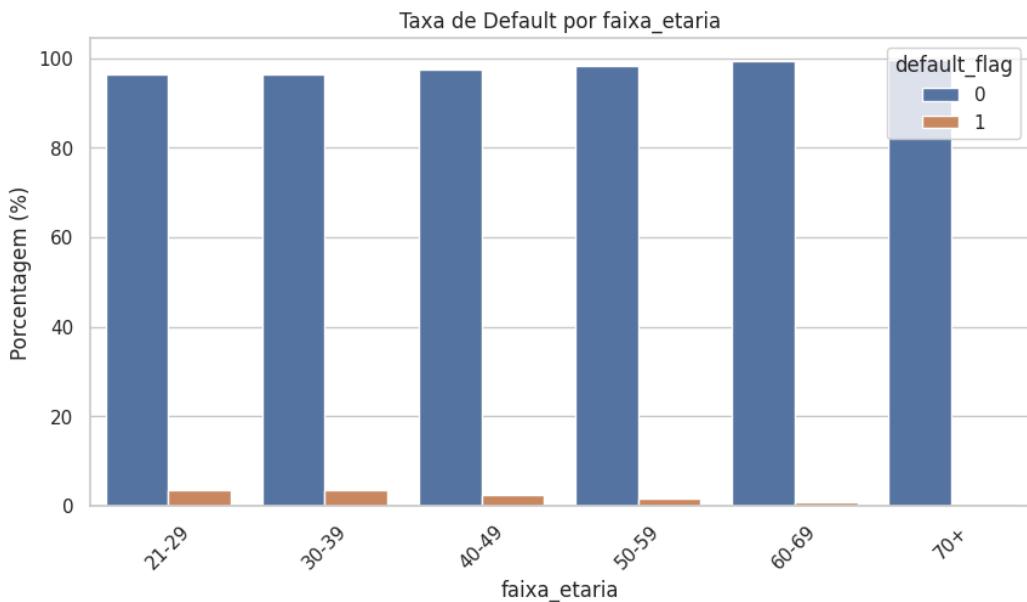
        sns.barplot(data=prop_df, x=col, y='percent', hue='default_flag')
        plt.title(f'Taxa de Default por {col}')
        plt.ylabel('Porcentagem (%)')

    plt.xticks(rotation=45)
    plt.show()

# Exemplo de uso
for col in ['age', 'last_month_salary', 'debt_ratio', 'faixa_etaria']:
    numeric = True if col in numeric_cols else False
    plot_target_analysis(col, numeric)
```



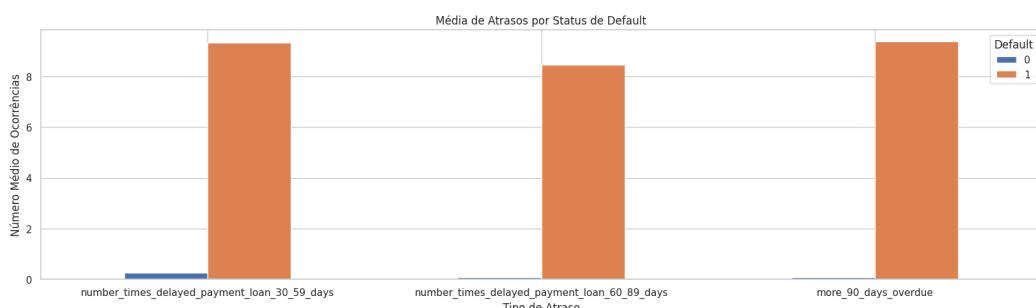




```
# Análise de Comportamento de Pagamento
# Criar um dataframe de métricas de pagamento
payment_metrics = df[['default_flag',
                      'number_times_delayed_payment_loan_30_59_days',
                      'number_times_delayed_payment_loan_60_89_days',
                      'more_90_days_overdue']]

# Agregar por status de default
payment_summary = payment_metrics.groupby('default_flag').mean().T

# Plotar
payment_summary.plot(kind='bar', figsize=(20, 5))
plt.title('Média de Atrasos por Status de Default')
plt.ylabel('Número Médio de Ocorrências')
plt.xlabel('Tipo de Atraso')
plt.xticks(rotation=0)
plt.legend(title='Default')
plt.show()
```



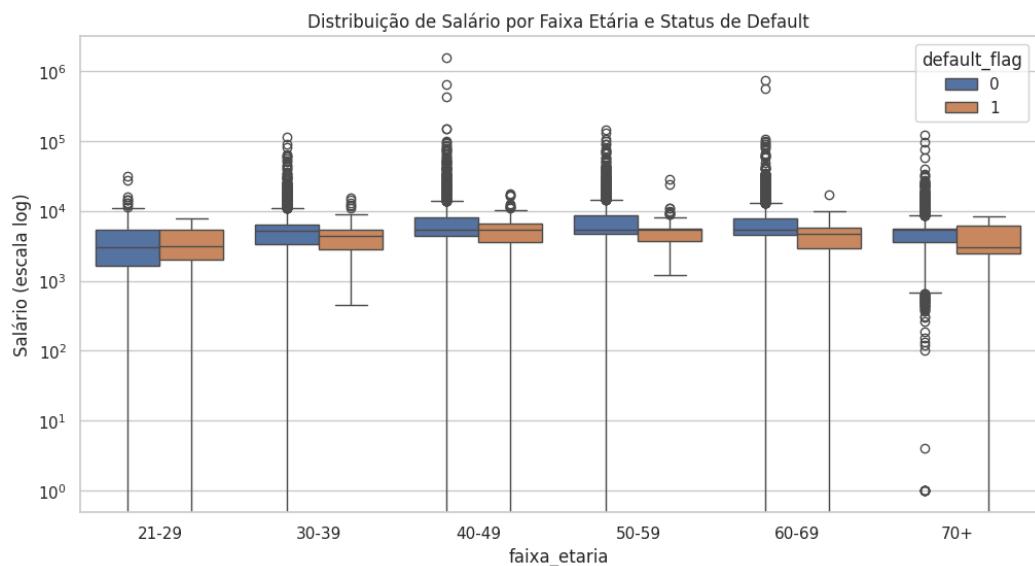
```
# Segmentação e Análise Multivariada
# Segmentação por faixa etária e análise de renda vs default
plt.figure(figsize=(12, 6))
```

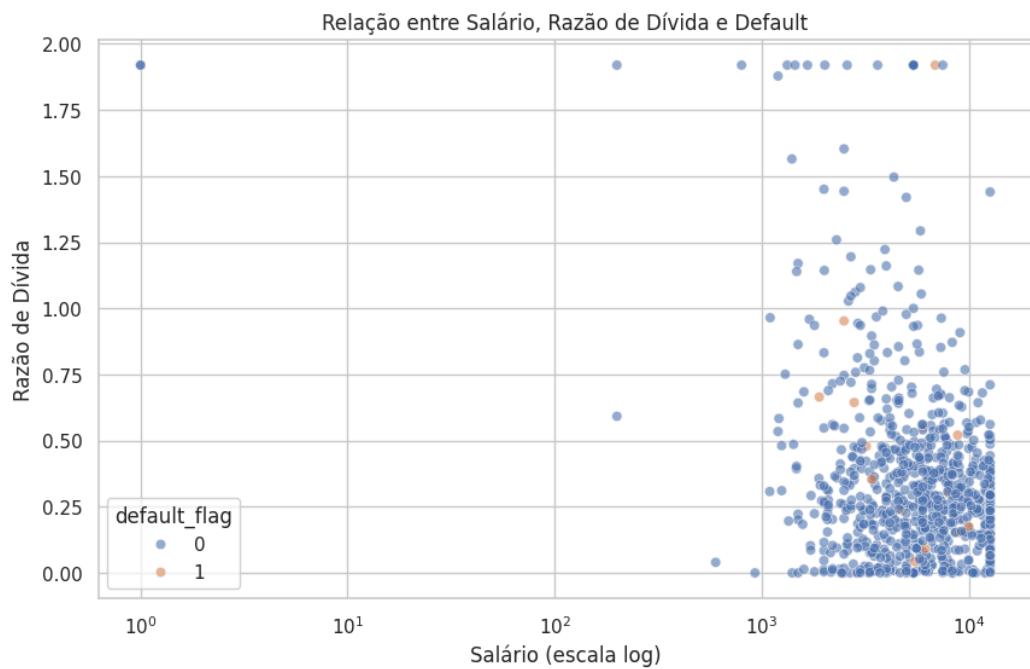
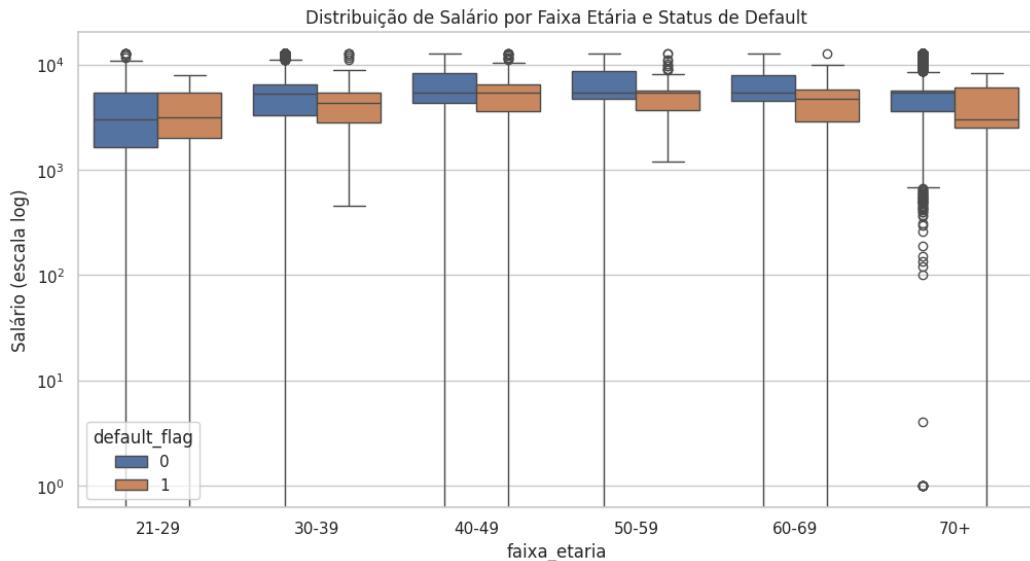
```

sns.boxplot(data=df, x='faixa_etaria', y='last_month_salary', hue='default_flag')
plt.title('Distribuição de Salário por Faixa Etária e Status de Default')
plt.yscale('log') # Usando escala log para melhor visualização
plt.ylabel('Salário (escala log)')
plt.show()

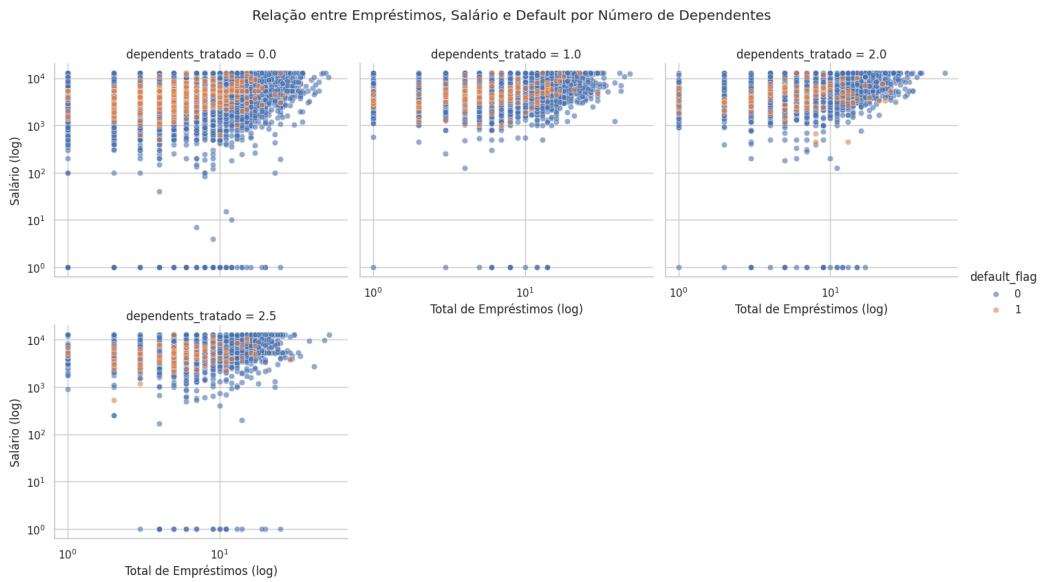
# Análise de dívida vs renda
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df.sample(1000), # Amostra para melhor visualização
                x='last_month_salary',
                y='debt_ratio',
                hue='default_flag',
                alpha=0.6)
plt.xscale('log')
plt.title('Relação entre Salário, Razão de Dívida e Default')
plt.xlabel('Salário (escala log)')
plt.ylabel('Razão de Dívida')
plt.show()

```





```
# Análise de Empréstimos e Dependents
# Criar grid de análise
g = sns.FacetGrid(data=df, col='number_dependents', hue='default_flag',
                   col_wrap=3, height=4, aspect=1.2)
g.map(sns.scatterplot, 'total_emprestimos', 'last_month_salary', alpha=0.6)
g.add_legend()
g.set(xscale='log', yscale='log')
g.set_axis_labels('Total de Empréstimos (log)', 'Salário (log)')
plt.suptitle('Relação entre Empréstimos, Salário e Default por Número de Dependentes', y=1.05)
plt.show()
```



```
# Identificação de Outliers
def detect_outliers(col):
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]

    print(f"\nAnálise de Outliers para {col}:")
    print(f"- Limite inferior: {lower_bound:.2f}")
    print(f"- Limite superior: {upper_bound:.2f}")
    print(f"- Número de outliers: {len(outliers)}")
    print(f"- Porcentagem de outliers: {len(outliers)/len(df)*100:.2f}%")

    return outliers

# Exemplo para variáveis chave
for col in ['last_month_salary', 'debt_ratio', 'age', 'total_emprestimos']:
    outliers = detect_outliers(col)
```

Análise de Outliers para salary\_tratado:

Limite inferior: -1374.00

- Limite superior: 12690.00
- Número de outliers: 0
- Porcentagem de outliers: 0.00%

Análise de Outliers para debt\_ratio\_tratado:

- Limite inferior: -0.87
- Limite superior: 1.92

- Número de outliers: 0
- Porcentagem de outliers: 0.00%

Análise de Outliers para age\_tratado:

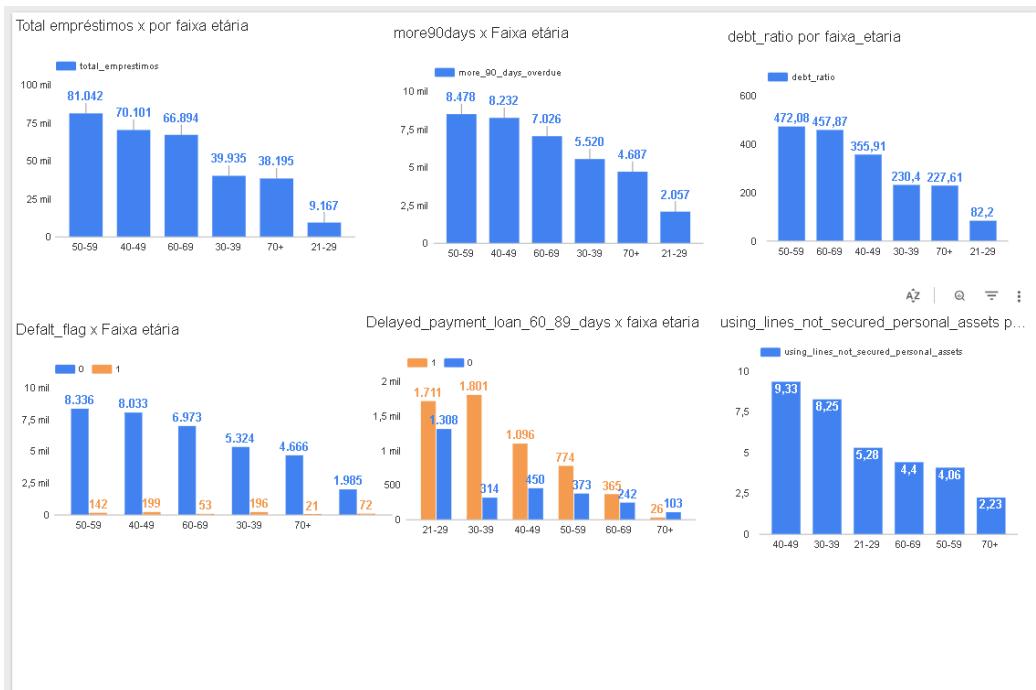
- Limite inferior: 8.00
- Limite superior: 96.00
- Número de outliers: 0
- Porcentagem de outliers: 0.00%

Análise de Outliers para total\_emprestimos:

- Limite inferior: -4.00
- Limite superior: 20.00
- Número de outliers: 975
- Porcentagem de outliers: 2.71%

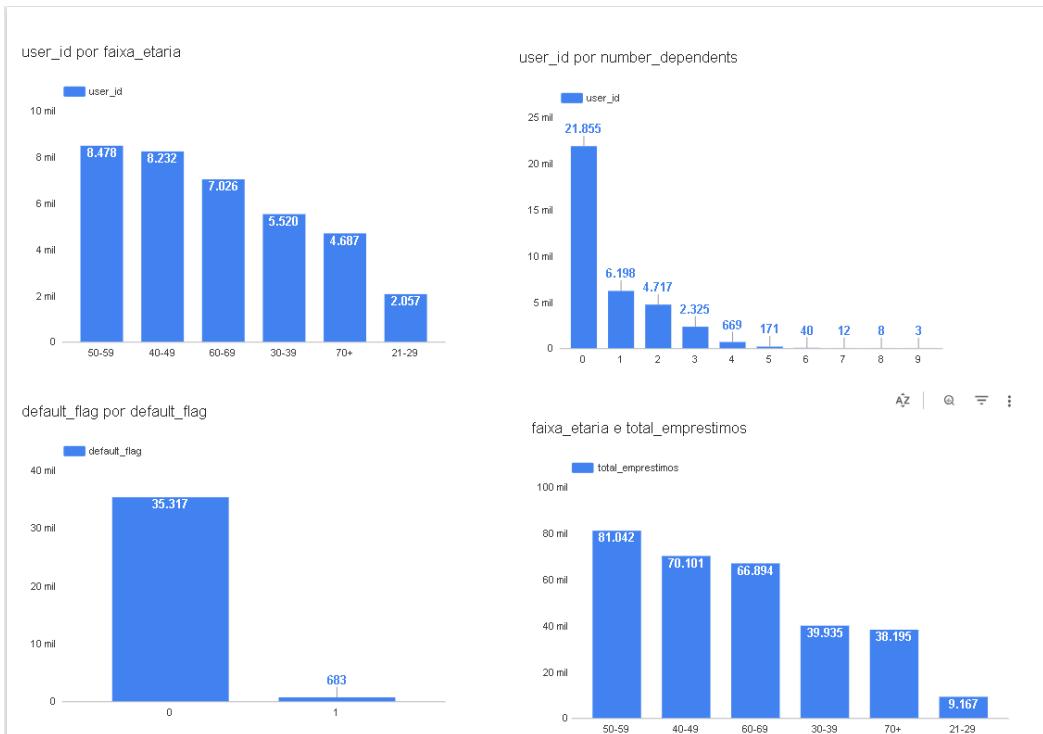


## 2.2.1 🔮 Agrupar dados de acordo com variáveis categóricas



## 2.2.2 🔮 Ver variáveis categóricas

A distribuição de variáveis categóricas como `faixa_etaria` e `default_flag` foi analisada usando gráficos de contagem para entender a composição da base de dados.



### 2.2.3 Aplicar medidas de tendência central (moda, média, mediana)

SELECT

```
-- Médias (AVG)
AVG(age_tratado) AS media_idade,
AVG(salary_tratado) AS media_salario,
AVG(dependents_tratado) AS media_dependentes,
AVG(more_90_days_overdue) AS media_atraso_90_dias,
AVG(using_lines_tratado) AS media_linhas_nao_garantidas,
AVG(number_times_delayed_payment_loan_30_59_days) AS media_atraso_30_59_dias,
AVG(debt_ratio_tratado) AS media_taxa_divida,
AVG(number_times_delayed_payment_loan_60_89_days) AS media_atraso_60_89_dias,
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
LIMIT 1;
SELECT
-- Medianas (PERCENTILE_CONT)
PERCENTILE_CONT(age_tratado, 0.5) OVER() AS mediana_idade,
PERCENTILE_CONT(salary_tratado, 0.5) OVER() AS mediana_salario,
PERCENTILE_CONT(dependents_tratado, 0.5) OVER() AS mediana_dependentes,
PERCENTILE_CONT(more_90_days_overdue, 0.5) OVER() AS mediana_atraso_90_dias,
PERCENTILE_CONT(using_lines_tratado, 0.5) OVER() AS mediana_linhas_nao_garantidas,
PERCENTILE_CONT(number_times_delayed_payment_loan_30_59_days, 0.5) OVER() AS mediana_atra
so_30_59_dias,
PERCENTILE_CONT(debt_ratio_tratado, 0.5) OVER() AS mediana_taxa_divida,
PERCENTILE_CONT(number_times_delayed_payment_loan_60_89_days, 0.5) OVER() AS mediana_atra
so_60_89_dias,
FROM
```

```

`my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
LIMIT 1;

SELECT
    -- Modas (usando subconsultas)
    (SELECT APPROX_TOP_COUNT(age_tratado, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_idade,
    (SELECT APPROX_TOP_COUNT(salary_tratado, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_salario,
    (SELECT APPROX_TOP_COUNT(dependents_tratado, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_dependentes,
    (SELECT APPROX_TOP_COUNT(more_90_days_overdue, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_atraso_90_dias,
    (SELECT APPROX_TOP_COUNT(using_lines_tratado, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_linhas_nao_garantidas,
    (SELECT APPROX_TOP_COUNT(number_times_delayed_payment_loan_30_59_days, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_atraso_30_59_dias,
    (SELECT APPROX_TOP_COUNT(debt_ratio_tratado, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_taxa_divida,
    (SELECT APPROX_TOP_COUNT(number_times_delayed_payment_loan_60_89_days, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_atraso_60_89_dias

FROM
    `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
LIMIT 1;

```

The screenshot shows the Google Cloud BigQuery interface. On the left, there's a sidebar with navigation links like 'BigQuery', 'Studio', 'Pipelines e integração', 'Transferências de dados', 'Dataform', 'Consultas programadas', 'Programação', 'Governança', 'Compartilhamento (Anônimo)', 'Tags de política', 'Curadoria de metadados...', 'Administração', 'Monitoramento', 'Central de parceiros', 'Configurações', and 'Notas de lançamento'. The main area has tabs for 'Consulta concluída' and 'Resultados da consulta'. The query results table has columns: 'Linha', 'media\_idade', 'media\_salario', 'media\_dependentes', 'media\_atraso\_30\_dias', 'media\_linhas\_nao\_garantidas', 'media\_atraso\_30\_59\_dias', 'media\_taxa\_divida', and 'media\_atraso\_60\_89\_dias'. One row of data is shown: Line 1 with values 52.41649999999999, 5885.8709722222, 0.6585277777777778, 0.2600855555555555, 0.318392644915, 0.4192777777777778, 0.667889519212, and 0.2378611111111111.

Linha	media_idade	media_salario	media_dependentes	media_atraso_30_dias	media_linhas_nao_garantidas	media_atraso_30_59_dias	media_taxa_divida	media_atraso_60_89_dias
1	52.41649999999999	5885.8709722222	0.6585277777777778	0.2600855555555555	0.318392644915	0.4192777777777778	0.667889519212	0.2378611111111111

```

2.2.3 Aplicar medidas de te...
11 FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
12
13 LIMIT 1;
14
15     Medidas ([PERCENTILE_CONT])
16     . . .
17     PERCENTILE_CONT(age_tratado, 0.5) OVER() AS mediana_idade,
18     PERCENTILE_CONT(salary_tratado, 0.5) OVER() AS mediana_salario,
19     PERCENTILE_CONT(dependents_tratado, 0.5) OVER() AS mediana_dependentes,
20     PERCENTILE_CONT(more_90_days_overdue, 0.5) OVER() AS mediana_atraso_90_dias,
21     PERCENTILE_CONT(number_lines_delayed_payment_loan_30_59_days, 0.5) OVER() AS mediana_atraso_30_59_dias,
22     PERCENTILE_CONT(debt_ratio_tratado, 0.5) OVER() AS mediana_taxa_divida,
23     PERCENTILE_CONT(number_times_delayed_payment_loan_60_89_days, 0.5) OVER() AS mediana_atraso_60_89_dias,
24
25 FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
26
27 LIMIT 1;
28
29 Modas (usando subconsultas)
30
31 Consulta concluída

```

**Resultados da consulta**

Linha	mediana_idade	mediana_salario	mediana_dependentes	mediana_atraso_90_dias	mediana_atraso_30_59_dias	mediana_taxa_divida	mediana_atraso_60_89_dias
1	52.0	5416.0	0.0	0.0	0.149654910500...	0.0	0.366168708500...

Resultados por página: 50 ▾ 1 - 1 de 1 < > ▾

```

2.2.3 Aplicar medidas de te...
31     (SELECT APPROX_TOP_COUNT(salary_tratado, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_salario,
32     (SELECT APPROX_TOP_COUNT(dependents_tratado, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_dependentes,
33     (SELECT APPROX_TOP_COUNT(more_90_days_overdue, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_atraso_90_dias,
34     (SELECT APPROX_TOP_COUNT(using_lines_tratado, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_linhas_nao_garantidas,
35     (SELECT APPROX_TOP_COUNT(number_lines_delayed_payment_loan_30_59_days, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_atraso_30_59_dias,
36     (SELECT APPROX_TOP_COUNT(debt_ratio_tratado, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_taxa_divida,
37     (SELECT APPROX_TOP_COUNT(number_times_delayed_payment_loan_60_89_days, 1)[OFFSET(0)].value FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`) AS moda_atraso_60_89_dias
38
39 FROM `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`;
40
41 Consulta concluída

```

**Resultados da consulta**

Linha	moda_idade	moda_salario	moda_dependentes	moda_atraso_90_dias	moda_atraso_30_59_dias	moda_taxa_divida	moda_atraso_60_89_dias	
1	50.0	5416.0	0.0	0	0.0	0	1.919481948749...	0

Resultados por página: 50 ▾ 1 - 1 de 1 < > ▾



## 2.2.4 Ver distribuição

As medidas de tendência central (média, mediana, moda) e a distribuição das variáveis numéricas foram calculadas usando SQL no BigQuery e visualizadas com histogramas e boxplots no Colab.

### 1. Análise com SQL

Em SQL, você pode obter as mesmas estatísticas usando funções de agregação como `COUNT`, `AVG`, `STDDEV` e `PERCENTILE_CONT`.

#### Código SQL:

SQL

```

SELECT
    COUNT(age_tratado) AS contagem_idade,
    AVG(age_tratado) AS media_idade,
    STDDEV(age_tratado) AS desvio_padrao_idade,
    MIN(age_tratado) AS min_idade,
    PERCENTILE_CONT(age_tratado, 0.25) OVER() AS q1_idade,

```

```

PERCENTILE_CONT(age_tratado, 0.5) OVER() AS mediana_idade,
PERCENTILE_CONT(age_tratado, 0.75) OVER() AS q3_idade,
MAX(age_tratado) AS max_idade,
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
GROUP BY age_tratado
LIMIT 1;

```

```

SELECT
COUNT(salary_tratado) AS contagem_salario,
AVG(salary_tratado) AS media_salario,
STDDEV(salary_tratado) AS desvio_padrao_salario,
MIN(salary_tratado) AS min_salario,
PERCENTILE_CONT(salary_tratado, 0.25) OVER() AS q1_salario,
PERCENTILE_CONT(salary_tratado, 0.5) OVER() AS mediana_salario,
PERCENTILE_CONT(salary_tratado, 0.75) OVER() AS q3_salario,
MAX(salary_tratado) AS max_salario
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
GROUP BY salary_tratado
LIMIT 1;

```

### O que os resultados te dirão:

- Histograma:** Ele mostra a frequência de cada faixa de valores. O `COUNT` e os valores de `min`, `max` e `q1` / `q3` te dão uma ideia de como os dados estão distribuídos.
- Boxplot:** O boxplot é construído com base nas estatísticas que o `describe()` e a consulta SQL fornecem: a mediana, o primeiro e o terceiro quartil, o valor mínimo e o máximo, e a detecção de outliers.

The screenshot shows the Google Cloud BigQuery interface. In the top navigation bar, the URL is `console.cloud.google.com/bigquery?project=my-project-laboratoria&dataset=dadoslaboratorioproject03&table=tb_quartis_compilada_com_outliers_tratados`. The main area displays a query editor with the following SQL code:

```

1 SELECT
2   COUNT(age_tratado) AS contagem_idade,
3   AVG(age_tratado) AS media_idade,
4   STDDEV(age_tratado) AS desvio_padrao_idade,
5   MIN(age_tratado) AS min_idade,
6   PERCENTILE_CONT(age_tratado, 0.25) OVER() AS q1_idade,
7   PERCENTILE_CONT(age_tratado, 0.5) OVER() AS mediana_idade,
8   PERCENTILE_CONT(age_tratado, 0.75) OVER() AS q3_idade,
9   MAX(age_tratado) AS max_idade,
10
11 FROM
12   `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
13 GROUP BY age_tratado
14 LIMIT 1;
15
16
17 SELECT

```

A note below the code says: "Este script processará 562,5 KB quando executado."

Below the code, there is a "Resultados da consulta" (Results) section. The results table has the following columns and data:

Linha	contagem_idade	media_idade	desvio_padrao_idade	min_idade	q1_idade	mediana_idade	q3_idade	max_idade
1	816	58.0	0.0	58.0	39.75	58.5	77.25	58.0

At the bottom of the results table, it says "Resultados por página: 50 1 - 1 de 1 < > >>"

Screenshot of the Google Cloud BigQuery interface showing a query script and its results.

```

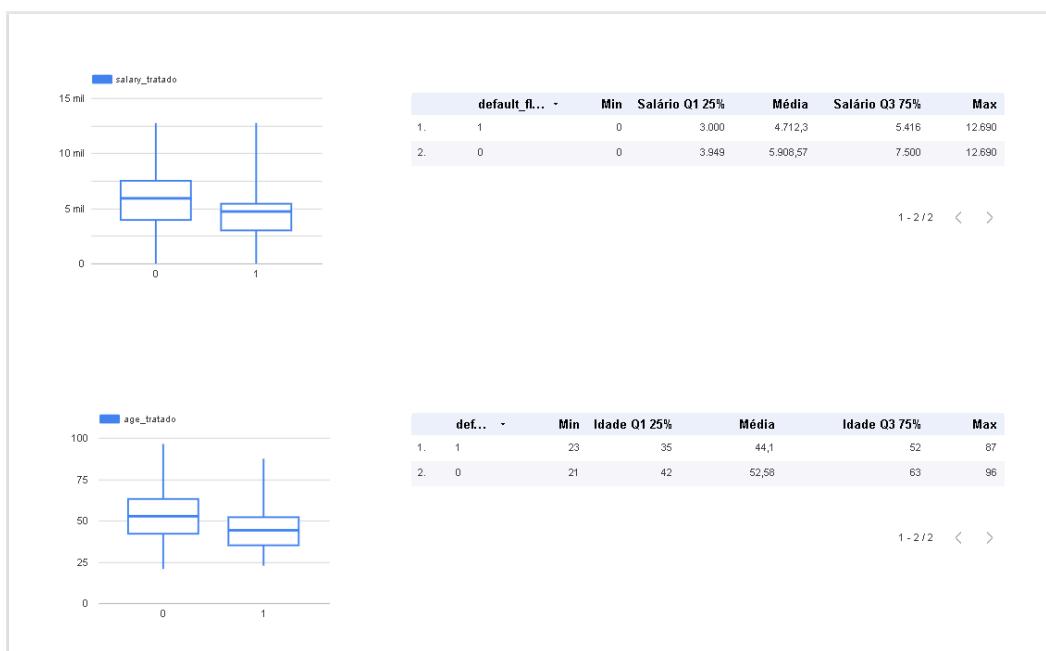
15
16
17 SELECT
18   COUNT(salary_tratado) AS contagem_salario,
19   AVG(salary_tratado) AS media_salario,
20   STDDEV(salary_tratado) AS desvio_padrao_salario,
21   MIN(salary_tratado) AS min_salario,
22   PERCENTILE_CONT(salary_tratado, 0.25) OVER() AS q1_salario,
23   PERCENTILE_CONT(salary_tratado, 0.50) OVER() AS mediana_salario,
24   PERCENTILE_CONT(salary_tratado, 0.75) OVER() AS q3_salario,
25   MAX(salary_tratado) AS max_salario
26
27 FROM
28   `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
29 GROUP BY salary_tratado
30
31

```

**Este script processará 562,5 KB quando executado.**

**Resultados da consulta**

Linha	contagem_salario	media_salario	desvio_padrao_salario	min_salario	q1_salario	mediana_salario	q3_salario	max_salario
1	2	8668,0	0,0	8668,0	3355,0	5527,0	8120,0	8668,0



## 2.2.5 Aplicar medidas de dispersão (desvio padrão)

O desvio padrão de variáveis como `age_tratado`, `salary_tratado`, e `debt_ratio_tratado` foi calculado para entender a dispersão dos dados.

2.2.5 Aplicar medidas de dispersão

```

1 #verificando desvio padrão
2 SELECT
3 STDDEV_SAMP(number_times_delayed_payment_loan_30_59_days) AS more_59 ,
4 STDDEV_SAMP(number_times_delayed_payment_loan_30_59_days) AS more_90,
5 STDDEV_SAMP(number_times_delayed_payment_loan_60_89_days) AS dias_89,
6 STDDEV_SAMP(age_tratado) AS age,
7 STDDEV_SAMP(debt_ratio_tratado) AS debt_ratio_tratado,
8 STDDEV_SAMP(dependents_tratado) AS dependents_tratado,
9 STDDEV_SAMP(salary_tratado) AS salary_tratado,
10 STDDEV_SAMP(using_lines_tratado) AS using_lines_tratado
11 FROM
12 `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
```

Esta consulta vai processar 2,2 MB quando executada.

Resultados da consulta

Linha	more_59	more_90	dias_89	age	debt_ratio_tratado	dependents_tratado	salary_tratado	using_lines_tratado
1	4.144020438225...	4.121364668426...	4.105514755101...	14.78839230946...	0.693962961258...	0.907417051273...	3050.977269599...	0.353819...

Project03lab

default_flag	desvio padrão age	desvio padrão debt_ratio	desvio padrão dependents_tratado	desvio padrão salary	desvio padrão more90...
1	14,79	0,89	0,91	3.098,0	1,83
2	12,09	0,87	0,98	2.966,02	2,25

default_flag	desvio padrão 30_59	desvio padrão 60_89	desvio padrão using_lines
1	25,24	25,47	0,23
2	1,91	1,82	0,34

## 2.2.6 Calcular quartis, decis ou percentis

Para o cálculo do risco relativo, a função `Ntile(4)` foi utilizada para dividir as variáveis numéricas em quatro grupos (quartis), o que permitiu a comparação de risco entre os grupos.

Os quartis foram feitos para as seguintes variáveis

- salary\_tratado
- dependents\_tratado
- age\_tratado
- using\_lines\_tratado
- number\_times\_delayed\_payment\_loan\_30\_59\_days
- number\_times\_delayed\_payment\_loan\_60\_89\_days
- more\_90\_days\_overdue
- debt\_ratio
- total\_emprestimos
- qtd\_imoveis\_emprestimos
- qtd\_outros\_emprestimos

```

CREATE OR REPLACE TABLE
`my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados` AS
WITH
base_outliers_tratados AS (
SELECT
user_id,
default_flag,
age,
faixa_etaria,
last_month_salary,
number_dependents,
total_emprestimos,
qtd_imoveis_emprestimos,
qtd_outros_emprestimos,
more_90_days_overdue,
using_lines_not_secured_personal_assets,
number_times_delayed_payment_loan_30_59_days,
number_times_delayed_payment_loan_60_89_days,
debt_ratio,
-- Cálculo dos quartis para o tratamento de outliers
PERCENTILE_CONT(age, 0.25) OVER() AS age_q1,
PERCENTILE_CONT(age, 0.75) OVER() AS age_q3,
PERCENTILE_CONT(last_month_salary, 0.25) OVER() AS salary_q1,
PERCENTILE_CONT(last_month_salary, 0.75) OVER() AS salary_q3,
PERCENTILE_CONT(number_dependents, 0.25) OVER() AS dependents_q1,
PERCENTILE_CONT(number_dependents, 0.75) OVER() AS dependents_q3,
PERCENTILE_CONT(using_lines_not_secured_personal_assets, 0.25) OVER() AS using_lines_q1,
PERCENTILE_CONT(using_lines_not_secured_personal_assets, 0.75) OVER() AS using_lines_q3,
PERCENTILE_CONT(debt_ratio, 0.25) OVER() AS debt_ratio_q1,
PERCENTILE_CONT(debt_ratio, 0.75) OVER() AS debt_ratio_q3
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_consolida_final` ),
base_com_valores_tratados AS (
SELECT
user_id,
default_flag,
age,
faixa_etaria,
last_month_salary,
number_dependents,
total_emprestimos,
qtd_imoveis_emprestimos,
qtd_outros_emprestimos,
more_90_days_overdue,
using_lines_not_secured_personal_assets,
number_times_delayed_payment_loan_30_59_days,
number_times_delayed_payment_loan_60_89_days,
debt_ratio,
-- Tratar outliers de idade
CASE
WHEN age < (age_q1 - 1.5 * (age_q3 - age_q1)) THEN (age_q1 - 1.5 * (age_q3 - age_q1))
WHEN age > (age_q3 + 1.5 * (age_q3 - age_q1)) THEN (age_q3 + 1.5 * (age_q3 - age_q1))

```

```

ELSE age
END
AS age_tratado,
-- Tratar outliers de salário
CASE
    WHEN last_month_salary < (salary_q1 - 1.5 * (salary_q3 - salary_q1)) THEN (salary_q1 - 1.5 * (salary_q3 - salary_q1))
    WHEN last_month_salary > (salary_q3 + 1.5 * (salary_q3 - salary_q1)) THEN (salary_q3 + 1.5 * (salary_q3 - salary_q1))
    ELSE last_month_salary
END
AS salary_tratado,
-- Tratar outliers de number_dependents
CASE
    WHEN number_dependents < (dependents_q1 - 1.5 * (dependents_q3 - dependents_q1)) THEN (dependents_q1 - 1.5 * (dependents_q3 - dependents_q1))
    WHEN number_dependents > (dependents_q3 + 1.5 * (dependents_q3 - dependents_q1)) THEN (dependents_q3 + 1.5 * (dependents_q3 - dependents_q1))
    ELSE number_dependents
END
AS dependents_tratado,
-- Tratar outliers de using_lines_not_secured_personal_assets
CASE
    WHEN using_lines_not_secured_personal_assets < (using_lines_q1 - 1.5 * (using_lines_q3 - using_lines_q1)) THEN (using_lines_q1 - 1.5 * (using_lines_q3 - using_lines_q1))
    WHEN using_lines_not_secured_personal_assets > (using_lines_q3 + 1.5 * (using_lines_q3 - using_lines_q1)) THEN (using_lines_q3 + 1.5 * (using_lines_q3 - using_lines_q1))
    ELSE using_lines_not_secured_personal_assets
END
AS using_lines_tratado,
-- Tratar outliers de debt_ratio
CASE
    WHEN debt_ratio < (debt_ratio_q1 - 1.5 * (debt_ratio_q3 - debt_ratio_q1)) THEN (debt_ratio_q1 - 1.5 * (debt_ratio_q3 - debt_ratio_q1))
    WHEN debt_ratio > (debt_ratio_q3 + 1.5 * (debt_ratio_q3 - debt_ratio_q1)) THEN (debt_ratio_q3 + 1.5 * (debt_ratio_q3 - debt_ratio_q1))
    ELSE debt_ratio
END
AS debt_ratio_tratado
FROM
base_outliers_tratados )
SELECT
user_id,
default_flag,
faixa_etaria,
total_emprestimos,
qtd_imoveis_emprestimos,
qtd_outros_emprestimos,
more_90_days_overdue,
number_times_delayed_payment_loan_30_59_days,
number_times_delayed_payment_loan_60_89_days,
age_tratado,

```

```

salary_tratado,
debt_ratio_tratado,
using_lines_tratado,
dependents_tratado,

-- Calcular os NTILEs com base nas variáveis numéricas
NTILE(4) OVER (ORDER BY age_tratado) AS age_quartil,
NTILE(4) OVER (ORDER BY salary_tratado) AS salary_quartil,
NTILE(4) OVER (ORDER BY dependents_tratado) AS dependent_quartil,
NTILE(4) OVER (ORDER BY more_90_days_overdue) AS more_90_days_quartil,
NTILE(4) OVER (ORDER BY using_lines_tratado) AS using_lines_quartil,
NTILE(4) OVER (ORDER BY number_times_delayed_payment_loan_30_59_days) AS delayed_payment_3
0_59_quartil,
NTILE(4) OVER (ORDER BY number_times_delayed_payment_loan_60_89_days) AS delayed_payment_6
0_89_quartil,
NTILE(4) OVER (ORDER BY debt_ratio_tratado) AS debt_ratio_quartil,
NTILE(4) OVER (ORDER BY total_emprestimos) AS qtd_emprestimos_total_quartil,
NTILE(4) OVER (ORDER BY qtd_imoveis_emprestimos) AS qtd_imoveis_emprestimos_quartil,
NTILE(4) OVER (ORDER BY qtd_outros_emprestimos) AS qtd_outros_emprestimos_quartil
FROM
base_com_valores_tratados;

```

The screenshot shows the Google Cloud BigQuery web interface. In the center, there is a query editor window with the following code:

```

93   more_90_days_overdue,
94   number_times_delayed_payment_loan_30_59_days,
95   number_times_delayed_payment_loan_60_89_days,
96   age_tratado,
97   salary_tratado,
98   debt_ratio_tratado,
99   using_lines_tratado,
100  dependents_tratado,
101
102  -- Calcular os NTILEs com base nas variáveis numéricas
103  NTILE(4) OVER (ORDER BY age_tratado) AS age_quartil,
104  NTILE(4) OVER (ORDER BY salary_tratado) AS salary_quartil,
105  NTILE(4) OVER (ORDER BY dependents_tratado) AS dependent_quartil,
106  NTILE(4) OVER (ORDER BY more_90_days_overdue) AS more_90_days_quartil,
107  NTILE(4) OVER (ORDER BY using_lines_tratado) AS using_lines_quartil,
108  NTILE(4) OVER (ORDER BY number_times_delayed_payment_loan_30_59_days) AS
delayed_payment_30_59_quartil,
109  NTILE(4) OVER (ORDER BY number_times_delayed_payment_loan_60_89_days) AS
delayed_payment_60_89_quartil,
110  NTILE(4) OVER (ORDER BY debt_ratio_tratado) AS debt_ratio_quartil,
111  NTILE(4) OVER (ORDER BY total_emprestimos) AS qtd_emprestimos_total_quartil,
112  NTILE(4) OVER (ORDER BY qtd_imoveis_emprestimos) AS qtd_imoveis_emprestimos_quartil,
113  NTILE(4) OVER (ORDER BY qtd_outros_emprestimos) AS qtd_outros_emprestimos_quartil
114  FROM
115  base_com_valores_tratados;

```

A red box highlights the identifier 'qtd\_outros\_emprestimos' at line 113, which is underlined with a red line. A tooltip below the box says: "Syntax error: Unexpected identifier 'qtd\_outros\_emprestimos' at [1:1]. Ativar o Windows".

### 2.2.7 Calcular correlação entre variáveis numéricas

A função `CORR` foi utilizada para analisar a correlação entre as variáveis e a `default_flag`.

```

SELECT
-- Correlação da idade com as outras variáveis
CORR(age_tratado, salary_tratado) AS age_salary_corr,
CORR(age_tratado, dependents_tratado) AS age_dependents_corr,
CORR(age_tratado, debt_ratio_tratado) AS age_debt_ratio_corr,
CORR(age_tratado, using_lines_tratado) AS age_using_lines_corr,
CORR(age_tratado, default_flag) AS age_default_flag_corr,

```

```

CORR(age_tratado, more_90_days_overdue) AS age_more_90_days_corr,
CORR(age_tratado, total_emprestimos) AS age_total_emprestimos_corr,
CORR(age_tratado, number_times_delayed_payment_loan_30_59_days) AS age_delay_30_59_corr,
CORR(age_tratado, number_times_delayed_payment_loan_60_89_days) AS age_delay_60_89_corr,

-- Correlação do salário com as outras variáveis
CORR(salary_tratado, dependents_tratado) AS salary_dependents_corr,
CORR(salary_tratado, debt_ratio_tratado) AS salary_debt_ratio_corr,
CORR(salary_tratado, using_lines_tratado) AS salary_using_lines_corr,
CORR(salary_tratado, default_flag) AS salary_default_flag_corr,
CORR(salary_tratado, more_90_days_overdue) AS salary_more_90_days_corr,
CORR(salary_tratado, total_emprestimos) AS salary_total_emprestimos_corr,
CORR(salary_tratado, number_times_delayed_payment_loan_30_59_days) AS salary_delay_30_59_corr,
CORR(salary_tratado, number_times_delayed_payment_loan_60_89_days) AS salary_delay_60_89_corr,

-- Correlação dos dependentes com as outras variáveis
CORR(dependents_tratado, debt_ratio_tratado) AS dependents_debt_ratio_corr,
CORR(dependents_tratado, using_lines_tratado) AS dependents_using_lines_corr,
CORR(dependents_tratado, default_flag) AS dependents_default_flag_corr,
CORR(dependents_tratado, more_90_days_overdue) AS dependents_more_90_days_corr,
CORR(dependents_tratado, total_emprestimos) AS dependents_total_emprestimos_corr,
CORR(dependents_tratado, number_times_delayed_payment_loan_30_59_days) AS dependents_delay_30_59_corr,
CORR(dependents_tratado, number_times_delayed_payment_loan_60_89_days) AS dependents_delay_60_89_corr,

-- Correlação da taxa de dívida com as outras variáveis
CORR(debt_ratio_tratado, using_lines_tratado) AS debt_ratio_using_lines_corr,
CORR(debt_ratio_tratado, default_flag) AS debt_ratio_default_flag_corr,
CORR(debt_ratio_tratado, more_90_days_overdue) AS debt_ratio_more_90_days_corr,
CORR(debt_ratio_tratado, total_emprestimos) AS debt_ratio_total_emprestimos_corr,
CORR(debt_ratio_tratado, number_times_delayed_payment_loan_30_59_days) AS debt_ratio_delay_30_59_corr,
CORR(debt_ratio_tratado, number_times_delayed_payment_loan_60_89_days) AS debt_ratio_delay_60_89_corr,

-- Correlação das linhas de crédito com as outras variáveis
CORR(using_lines_tratado, default_flag) AS using_lines_default_flag_corr,
CORR(using_lines_tratado, more_90_days_overdue) AS using_lines_more_90_days_corr,
CORR(using_lines_tratado, total_emprestimos) AS using_lines_total_emprestimos_corr,
CORR(using_lines_tratado, number_times_delayed_payment_loan_30_59_days) AS using_lines_delay_30_59_corr,
CORR(using_lines_tratado, number_times_delayed_payment_loan_60_89_days) AS using_lines_delay_60_89_corr,

-- Correlação do default_flag com as variáveis de atraso
CORR(default_flag, more_90_days_overdue) AS default_flag_more_90_days_corr,
CORR(default_flag, number_times_delayed_payment_loan_30_59_days) AS default_flag_delay_30_59_corr,
CORR(default_flag, number_times_delayed_payment_loan_60_89_days) AS default_flag_delay_60_89_corr,
CORR(default_flag, total_emprestimos) AS default_flag_total_emprestimos_corr

```

```
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados`
LIMIT 1;
```

```
[{
  "age_salary_corr": "0.091410424800549353",
  "age_dependents_corr": "-0.22944884848547828",
  "age_debt_ratio_corr": "0.085113556788871711",
  "age_using_lines_corr": "-0.28190335452246507",
  "age_default_flag_corr": "-0.078224625728620253",
  "age_more_90_days_corr": "-0.059921936780202967",
  "age_total_emprestimos_corr": "0.14511961308901042",
  "age_delay_30_59_corr": "-0.062559303214564835",
  "age_delay_60_89_corr": "-0.0563484287149074",
  "salary_dependents_corr": "0.20924965785439284",
  "salary_debt_ratio_corr": "-0.19859917331773236",
  "salary_using_lines_corr": "-0.0925754597230294",
  "salary_default_flag_corr": "-0.05349293650054425",
  "salary_more_90_days_corr": "-0.032039637276788123",
  "salary_total_emprestimos_corr": "0.26423056284961094",
  "salary_delay_30_59_corr": "-0.026325802717628639",
  "salary_delay_60_89_corr": "-0.027505346255223325",
  "dependents_debt_ratio_corr": "-0.16118107950388344",
  "dependents_using_lines_corr": "0.091877657349579839",
  "dependents_default_flag_corr": "0.030454984353217713",
  "dependents_more_90_days_corr": "-0.010676882242055413",
  "dependents_total_emprestimos_corr": "0.083599981755959216",
  "dependents_delay_30_59_corr": "-0.0037698308326353652",
  "dependents_delay_60_89_corr": "-0.011337409298546873",
  "debt_ratio_using_lines_corr": "-0.020493256495540849",
  "debt_ratio_default_flag_corr": "0.0039694909882166127",
  "debt_ratio_more_90_days_corr": "-0.020132582372620406",
  "debt_ratio_total_emprestimos_corr": "0.040327358579391542",
  "debt_ratio_delay_30_59_corr": "-0.01870463427491844",
  "debt_ratio_delay_60_89_corr": "-0.019920468909089908",
  "using_lines_default_flag_corr": "0.25161905875934815",
  "using_lines_more_90_days_corr": "0.10967997382091584",
  "using_lines_total_emprestimos_corr": "-0.15422780411323767",
  "using_lines_delay_30_59_corr": "0.12107878990179997",
  "using_lines_delay_60_89_corr": "0.097076865876859916",
  "default_flag_more_90_days_corr": "0.30748450335122296",
  "default_flag_delay_30_59_corr": "0.29920788112363939",
  "default_flag_delay_60_89_corr": "0.27825366272045821",
  "default_flag_total_emprestimos_corr": "-0.058156879828336505"
}]
```

## 1. Correlações com a `default_flag`

A correlação mais importante para o seu projeto é com a variável `default_flag`, pois ela indica o risco de inadimplência.

- **Correlação mais forte:** As variáveis de atraso de pagamento têm as maiores correlações com a `default_flag`.
  - `default_flag_more_90_days_corr` : **0,307**
  - `default_flag_delay_30_59_corr` : **0,299**
  - `default_flag_delay_60_89_corr` : **0,278**

Isso mostra que, como esperado, o histórico de atrasos é o principal fator que se correlaciona com a inadimplência. Quanto mais atrasos, maior o risco.
- **Correlação moderada:**
  - `using_lines_default_flag_corr` : **0,251**

Isso sugere que clientes que utilizam uma alta porcentagem de suas linhas de crédito não garantidas têm uma correlação moderada com a inadimplência. Faz sentido, pois um alto uso de crédito pode ser um sinal de risco financeiro.

## 2. Outras Correlações Relevantes

- **Correlação entre atrasos:**
  - `default_flag_more_90_days_corr` e `default_flag_delay_30_59_corr` (e outras combinações)
  - As variáveis de atraso de pagamento se correlacionam entre si, o que é esperado. Quem atrasa o pagamento tende a atrasar em diferentes períodos.
- **Correlação de Salário e Idade:**
  - `salary_total_emprestimos_corr` : **0,264** (correlação positiva moderada)
  - Isso indica que, em geral, clientes com salários mais altos tendem a ter um maior número de empréstimos.
  - `age_dependents_corr` : **0,229** (correlação negativa moderada)
  - Pessoas mais velhas tendem a ter um número menor de dependentes.
- **Correlações fracas:**
  - `debt_ratio_default_flag_corr` : **0,003**
  - `dependents_default_flag_corr` : **0,030**
  - Variáveis como a **taxa de endividamento** e o **número de dependentes** mostraram uma correlação muito fraca ou insignificante com a inadimplência.



### 2.3 Aplicar técnica de análise

```
WITH
proporcoes_gerais AS (
SELECT
COUNT(user_id) AS total_clientes,
SUM(default_flag) AS total_inadimplentes,
SUM(default_flag) / COUNT(user_id) AS proporcao_inadimplencia_geral
FROM
my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados
),
proporcoes_por_grupo AS (
```

```

SELECT
age_quartil,
COUNT(user_id) AS total_clientes_no_grupo,
SUM(default_flag) AS total_inadimplentes_no_grupo,
SUM(default_flag) / COUNT(user_id) AS proporcao_inadimplencia_no_grupo
FROM
my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados
GROUP BY
age_quartil
)
SELECT
p_grupo.age_quartil,
p_grupo.total_clientes_no_grupo,
p_grupo.total_inadimplentes_no_grupo,
p_grupo.proporcao_inadimplencia_no_grupo,
p_gerais.proporcao_inadimplencia_geral,
p_grupo.proporcao_inadimplencia_no_grupo / p_gerais.proporcao_inadimplencia_geral AS risco_relativo
FROM
proporcoes_por_grupo AS p_grupo
JOIN
proporcoes_gerais AS p_gerais
ON TRUE
ORDER BY
risco_relativo DESC;

```

Linha	age_quartil	total_clientes_no_grupo	total_inadimplentes_no_grupo	proporcao_inadimplencia_no_grupo	proporcao_inadimplencia_geral	risco_relativo
1	1	9000	309	0.0343333333333334	0.0189722222222222	1.809663250366...
2	2	9000	203	0.0225555555555555	0.0189722222222222	1.188872620790...
3	3	9000	121	0.0134444444444444	0.0189722222222222	0.70638360175...
4	4	9000	50	0.0055555555555555	0.0189722222222222	0.292825768667...

### 2.3.1 Calcular risco relativo

## 1. O que é Risco Relativo?

O **Risco Relativo** é uma métrica que compara a taxa de inadimplência de um grupo específico (um quartil de idade, por exemplo) com a taxa de inadimplência total de toda a base de clientes.

A fórmula usada é:

Risco Relativo = Taxa de Inadimplência Total / Taxa de Inadimplência do Grupo

## 2. Como o SQL realiza o cálculo?

O código usa "Tabelas de Expressão Comum" ( `WITH` ) para dividir a lógica em etapas fáceis de entender:

- `total_default_rate` : A primeira etapa calcula a **taxa de inadimplência total** da base de dados. Este valor será usado como o "denominador" para todos os cálculos de risco relativo. É a taxa de inadimplência média.
- `quartil_default_rates` : Esta é a etapa mais extensa. Ela calcula a **taxa de inadimplência individual** para cada um dos quartis de cada variável (idade, salário, etc.). O `UNION ALL` junta os resultados de todos os quartis em uma única tabela temporária.
- **Final `SELECT`** : Na etapa final, a consulta junta ( `CROSS JOIN` ) as duas tabelas temporárias. Para cada quartil, ela divide sua **taxa de inadimplência individual** (o **risco exposto**) pela **taxa de inadimplência total** (o **risco não exposto**), gerando o **risco relativo**.

```
-- Primeiro: Criar tabela com cálculo de risco relativo
CREATE OR REPLACE TABLE
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS
WITH
-- 1. Definir a base de dados principal
base_data AS (
SELECT
    user_id,
    default_flag,
    age_quartil,
    salary_quartil,
    dependent_quartil,
    more_90_days_quartil,
    using_lines_quartil,
    delayed_payment_30_59_quartil,
    delayed_payment_60_89_quartil,
    debt_ratio_quartil,
    qtd_emprestimos_total_quartil,
    qtd_imoveis_emprestimos_quartil,
    qtd_outros_emprestimos_quartil
FROM
    `my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados` ),
-- 2. Calcular a taxa de inadimplência total da base
total_default_rate AS (
SELECT
    SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_total
FROM
    base_data ),
-- 3. Calcular a taxa de inadimplência para cada quartil de cada variável
quartil_default_rates AS (
-- age_quartil
SELECT
    'age_quartil' AS tipo_variavel,
    CAST(age_quartil AS STRING) AS categoria,
    SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
    COUNT(*) AS total_clientes,
    COUNTIF(default_flag = 1) AS total_inadimplentes
FROM
```

```

base_data
GROUP BY
age_quartil
UNION ALL
-- salary_quartil
SELECT
'salary_quartil' AS tipo_variavel,
CAST(salary_quartil AS STRING) AS categoria,
SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
COUNT(*) AS total_clientes,
COUNTIF(default_flag = 1) AS total_inadimplentes
FROM
base_data
GROUP BY
salary_quartil
UNION ALL
-- dependent_quartil
SELECT
'dependent_quartil' AS tipo_variavel,
CAST(dependent_quartil AS STRING) AS categoria,
SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
COUNT(*) AS total_clientes,
COUNTIF(default_flag = 1) AS total_inadimplentes
FROM
base_data
GROUP BY
dependent_quartil
UNION ALL
-- more_90_days_quartil
SELECT
'more_90_days_quartil' AS tipo_variavel,
CAST(more_90_days_quartil AS STRING) AS categoria,
SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
COUNT(*) AS total_clientes,
COUNTIF(default_flag = 1) AS total_inadimplentes
FROM
base_data
GROUP BY
more_90_days_quartil
UNION ALL
-- using_lines_quartil
SELECT
'useing_lines_quartil' AS tipo_variavel,
CAST(using_lines_quartil AS STRING) AS categoria,
SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
COUNT(*) AS total_clientes,
COUNTIF(default_flag = 1) AS total_inadimplentes
FROM
base_data
GROUP BY
using_lines_quartil
UNION ALL

```

```

-- delayed_payment_30_59_quartil
SELECT
    'delayed_payment_30_59_quartil' AS tipo_variavel,
    CAST(delayed_payment_30_59_quartil AS STRING) AS categoria,
    SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
    COUNT(*) AS total_clientes,
    COUNTIF(default_flag = 1) AS total_inadimplentes
FROM
    base_data
GROUP BY
    delayed_payment_30_59_quartil
UNION ALL
-- delayed_payment_60_89_quartil
SELECT
    'delayed_payment_60_89_quartil' AS tipo_variavel,
    CAST(delayed_payment_60_89_quartil AS STRING) AS categoria,
    SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
    COUNT(*) AS total_clientes,
    COUNTIF(default_flag = 1) AS total_inadimplentes
FROM
    base_data
GROUP BY
    delayed_payment_60_89_quartil
UNION ALL
-- debt_ratio_quartil
SELECT
    'debt_ratio_quartil' AS tipo_variavel,
    CAST(debt_ratio_quartil AS STRING) AS categoria,
    SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
    COUNT(*) AS total_clientes,
    COUNTIF(default_flag = 1) AS total_inadimplentes
FROM
    base_data
GROUP BY
    debt_ratio_quartil
UNION ALL
-- qtd_emprestimos_total_quartil
SELECT
    'qtd_emprestimos_total_quartil' AS tipo_variavel,
    CAST(qtd_emprestimos_total_quartil AS STRING) AS categoria,
    SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
    COUNT(*) AS total_clientes,
    COUNTIF(default_flag = 1) AS total_inadimplentes
FROM
    base_data
GROUP BY
    qtd_emprestimos_total_quartil

UNION ALL
-- qtd_outros_emprestimos_quartil
SELECT
    'qtd_outros_emprestimos_quartil' AS tipo_variavel,

```

```

        CAST(qtd_outros_emprestimos_quartil AS STRING) AS categoria,
        SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
        COUNT(*) AS total_clientes,
        COUNTIF(default_flag = 1) AS total_inadimplentes
    FROM
        base_data
    GROUP BY
        qtd_outros_emprestimos_quartil

    UNION ALL
    -- qtd_imoveis_emprestimos_quartil
    SELECT
        'qtd_imoveis_emprestimos_quartil' AS tipo_variavel,
        CAST(qtd_imoveis_emprestimos_quartil AS STRING) AS categoria,
        SAFE_DIVIDE(COUNTIF(default_flag = 1), COUNT(*)) AS taxa_inadimplencia_quartil,
        COUNT(*) AS total_clientes,
        COUNTIF(default_flag = 1) AS total_inadimplentes
    FROM
        base_data
    GROUP BY
        qtd_imoveis_emprestimos_quartil
    )

-- 4. Unir os resultados e calcular o risco relativo final
SELECT
    t1.tipo_variavel,
    t1.categoria,
    -- A taxa de inadimplência da categoria é o "risco_exposto"
    t1.taxa_inadimplencia_quartil AS risco_exposto,
    -- A taxa de inadimplência total é o "risco_nao_exposto"
    t2.taxa_inadimplencia_total AS risco_nao_exposto,
    SAFE_DIVIDE(t1.taxa_inadimplencia_quartil, t2.taxa_inadimplencia_total) AS risco_relativo,
    t1.total_clientes,
    t1.total_inadimplentes
FROM
    quartil_default_rates AS t1
CROSS JOIN
    total_default_rate AS t2
ORDER BY
    tipo_variavel,
    risco_relativo DESC;

```

### 3. Interpretando os Resultados

- **Risco Relativo > 1:** Significa que o grupo tem um risco de inadimplência **maior que a média**.
- **Risco Relativo < 1:** Significa que o grupo tem um risco de inadimplência **menor que a média**.
- **Risco Relativo = 1:** O risco do grupo é **igual** ao risco médio da base.

```
[{
    "tipo_variavel": "age_quartil",
    "categoria": "1",
```

```

"risco_exposto": "0.03422222222222223",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "1.8038067349926792",
"total_clientes": "9000",
"total_inadimplentes": "308"
}, {
"tipo_variavel": "age_quartil",
"categoria": "2",
"risco_exposto": "0.0231111111111111",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "1.2181551976573937",
"total_clientes": "9000",
"total_inadimplentes": "208"
}, {
"tipo_variavel": "age_quartil",
"categoria": "3",
"risco_exposto": "0.0131111111111112",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "0.69106881405563692",
"total_clientes": "9000",
"total_inadimplentes": "118"
}, {
"tipo_variavel": "age_quartil",
"categoria": "4",
"risco_exposto": "0.005444444444444445",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "0.28696925329428985",
"total_clientes": "9000",
"total_inadimplentes": "49"
}, {
"tipo_variavel": "debt_ratio_quartil",
"categoria": "3",
"risco_exposto": "0.022444444444444444",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "1.1830161054172765",
"total_clientes": "9000",
"total_inadimplentes": "202"
}, {
"tipo_variavel": "debt_ratio_quartil",
"categoria": "4",
"risco_exposto": "0.01988888888888889",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "1.048316251830161",
"total_clientes": "9000",
"total_inadimplentes": "179"
}, {
"tipo_variavel": "debt_ratio_quartil",
"categoria": "1",
"risco_exposto": "0.01777777777777778",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "0.93704245973645672",
"total_clientes": "9000",

```

```

"total_inadimplentes": "160"
}, {
"tipo_variavel": "debt_ratio_quartil",
"categoria": "2",
"risco_exposto": "0.01577777777777779",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "0.83162518301610544",
"total_clientes": "9000",
"total_inadimplentes": "142"
}, {
"tipo_variavel": "delayed_payment_30_59_quartil",
"categoria": "4",
"risco_exposto": "0.06888888888888888888",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "3.63103953147877",
"total_clientes": "9000",
"total_inadimplentes": "620"
}, {
"tipo_variavel": "delayed_payment_30_59_quartil",
"categoria": "3",
"risco_exposto": "0.003",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "0.15812591508052709",
"total_clientes": "9000",
"total_inadimplentes": "27"
}, {
"tipo_variavel": "delayed_payment_30_59_quartil",
"categoria": "2",
"risco_exposto": "0.00288888888888888888",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "0.15226939970717421",
"total_clientes": "9000",
"total_inadimplentes": "26"
}, {
"tipo_variavel": "delayed_payment_30_59_quartil",
"categoria": "1",
"risco_exposto": "0.001111111111111111",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "0.058565153733528545",
"total_clientes": "9000",
"total_inadimplentes": "10"
}, {
"tipo_variavel": "delayed_payment_60_89_quartil",
"categoria": "4",
"risco_exposto": "0.06322222222222221",
"risco_nao_exposto": "0.01897222222222224",
"risco_relativo": "3.3323572474377743",
"total_clientes": "9000",
"total_inadimplentes": "569"
}, {
"tipo_variavel": "delayed_payment_60_89_quartil",
"categoria": "3",

```

```
"risco_exposto": "0.00444444444444444444",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.23426061493411418",
"total_clientes": "9000",
"total_inadimplentes": "40"
}, {
"tipo_variavel": "delayed_payment_60_89_quartil",
"categoria": "1",
"risco_exposto": "0.004111111111111114",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.21669106881405564",
"total_clientes": "9000",
"total_inadimplentes": "37"
}, {
"tipo_variavel": "delayed_payment_60_89_quartil",
"categoria": "2",
"risco_exposto": "0.004111111111111114",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.21669106881405564",
"total_clientes": "9000",
"total_inadimplentes": "37"
}, {
"tipo_variavel": "dependent_quartil",
"categoria": "4",
"risco_exposto": "0.025555555555555557",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "1.3469985358711567",
"total_clientes": "9000",
"total_inadimplentes": "230"
}, {
"tipo_variavel": "dependent_quartil",
"categoria": "2",
"risco_exposto": "0.018",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.94875549048316232",
"total_clientes": "9000",
"total_inadimplentes": "162"
}, {
"tipo_variavel": "dependent_quartil",
"categoria": "3",
"risco_exposto": "0.01788888888888888",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.94289897510980958",
"total_clientes": "9000",
"total_inadimplentes": "161"
}, {
"tipo_variavel": "dependent_quartil",
"categoria": "1",
"risco_exposto": "0.014444444444444444",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.76134699853587107",
"total_clientes": "9000",
```

```
"total_inadimplentes": "130"
}, {
  "tipo_variavel": "more_90_days_quartil",
  "categoria": "4",
  "risco_exposto": "0.07077777777777773",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "3.730600292825768",
  "total_clientes": "9000",
  "total_inadimplentes": "637"
}, {
  "tipo_variavel": "more_90_days_quartil",
  "categoria": "2",
  "risco_exposto": "0.001777777777777779",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "0.093704245973645683",
  "total_clientes": "9000",
  "total_inadimplentes": "16"
}, {
  "tipo_variavel": "more_90_days_quartil",
  "categoria": "1",
  "risco_exposto": "0.001666666666666668",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "0.087847730600292828",
  "total_clientes": "9000",
  "total_inadimplentes": "15"
}, {
  "tipo_variavel": "more_90_days_quartil",
  "categoria": "3",
  "risco_exposto": "0.001666666666666668",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "0.087847730600292828",
  "total_clientes": "9000",
  "total_inadimplentes": "15"
}, {
  "tipo_variavel": "qtd_emprestimos_total_quartil",
  "categoria": "1",
  "risco_exposto": "0.03411111111111113",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "1.7979502196193264",
  "total_clientes": "9000",
  "total_inadimplentes": "307"
}, {
  "tipo_variavel": "qtd_emprestimos_total_quartil",
  "categoria": "2",
  "risco_exposto": "0.018",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "0.94875549048316232",
  "total_clientes": "9000",
  "total_inadimplentes": "162"
}, {
  "tipo_variavel": "qtd_emprestimos_total_quartil",
  "categoria": "3",
```

```

"risco_exposto": "0.011888888888888888",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.6266471449487554",
"total_clientes": "9000",
"total_inadimplentes": "107"
}, {
"tipo_variavel": "qtd_emprestimos_total_quartil",
"categoria": "4",
"risco_exposto": "0.011888888888888888",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.6266471449487554",
"total_clientes": "9000",
"total_inadimplentes": "107"
}, {
"tipo_variavel": "qtd_imoveis_emprestimos_quartil",
"categoria": "1",
"risco_exposto": "0.029888888888888888",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "1.5754026354319179",
"total_clientes": "9000",
"total_inadimplentes": "269"
}, {
"tipo_variavel": "qtd_imoveis_emprestimos_quartil",
"categoria": "2",
"risco_exposto": "0.019666666666666666",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "1.0366032210834553",
"total_clientes": "9000",
"total_inadimplentes": "177"
}, {
"tipo_variavel": "qtd_imoveis_emprestimos_quartil",
"categoria": "3",
"risco_exposto": "0.013555555555555555",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.71449487554904823",
"total_clientes": "9000",
"total_inadimplentes": "122"
}, {
"tipo_variavel": "qtd_imoveis_emprestimos_quartil",
"categoria": "4",
"risco_exposto": "0.012777777777777779",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.67349926793557835",
"total_clientes": "9000",
"total_inadimplentes": "115"
}, {
"tipo_variavel": "qtd_outros_emprestimos_quartil",
"categoria": "1",
"risco_exposto": "0.033777777777777775",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "1.7803806734992675",
"total_clientes": "9000",

```

```
"total_inadimplentes": "304"
}, {
  "tipo_variavel": "qtd_outros_emprestimos_quartil",
  "categoria": "2",
  "risco_exposto": "0.01722222222222222",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "0.90775988286969245",
  "total_clientes": "9000",
  "total_inadimplentes": "155"
}, {
  "tipo_variavel": "qtd_outros_emprestimos_quartil",
  "categoria": "3",
  "risco_exposto": "0.012555555555555556",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "0.66178623718887264",
  "total_clientes": "9000",
  "total_inadimplentes": "113"
}, {
  "tipo_variavel": "qtd_outros_emprestimos_quartil",
  "categoria": "4",
  "risco_exposto": "0.01233333333333333",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "0.65007320644216682",
  "total_clientes": "9000",
  "total_inadimplentes": "111"
}, {
  "tipo_variavel": "salary_quartil",
  "categoria": "1",
  "risco_exposto": "0.029777777777777778",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "1.569546120058565",
  "total_clientes": "9000",
  "total_inadimplentes": "268"
}, {
  "tipo_variavel": "salary_quartil",
  "categoria": "2",
  "risco_exposto": "0.02011111111111111",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "1.0600292825768667",
  "total_clientes": "9000",
  "total_inadimplentes": "181"
}, {
  "tipo_variavel": "salary_quartil",
  "categoria": "3",
  "risco_exposto": "0.017666666666666667",
  "risco_nao_exposto": "0.01897222222222224",
  "risco_relativo": "0.93118594436310387",
  "total_clientes": "9000",
  "total_inadimplentes": "159"
}, {
  "tipo_variavel": "salary_quartil",
  "categoria": "4",
```

```

"risco_exposto": "0.008333333333333332",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.43923865300146409",
"total_clientes": "9000",
"total_inadimplentes": "75"
}, {
"tipo_variavel": "using_lines_quartil",
"categoria": "4",
"risco_exposto": "0.0711111111111111",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "3.7481698389458269",
"total_clientes": "9000",
"total_inadimplentes": "640"
}, {
"tipo_variavel": "using_lines_quartil",
"categoria": "3",
"risco_exposto": "0.003777777777777779",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.19912152269399708",
"total_clientes": "9000",
"total_inadimplentes": "34"
}, {
"tipo_variavel": "using_lines_quartil",
"categoria": "1",
"risco_exposto": "0.0008888888888888893",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.046852122986822842",
"total_clientes": "9000",
"total_inadimplentes": "8"
}, {
"tipo_variavel": "using_lines_quartil",
"categoria": "2",
"risco_exposto": "0.00011111111111111112",
"risco_nao_exposto": "0.018972222222222224",
"risco_relativo": "0.0058565153733528552",
"total_clientes": "9000",
"total_inadimplentes": "1"
}]

```



### 2.3.2 🔥 Aplicar segmentação por Score

Nesta etapa, o objetivo é utilizar a pontuação de risco para classificar os clientes. A abordagem do projeto, que utiliza variáveis dummy, foi implementada para criar um modelo de segmentação transparente e baseado em regras.

**Código SQL:** [tb\\_score\\_dummy.sql](#)

Este script realiza o seguinte:

- Criação de Variáveis Dummy:** Ele usa a tabela `tb_risco_relativo_completo` para criar variáveis binárias (0 ou 1) para cada quartil de cada variável, baseando-se em um ponto de corte (risco relativo > 1.2).

2. **Cálculo do Score:** A coluna `score_risco_total` é calculada como a soma de todas as variáveis dummy. Quanto maior a pontuação, maior o risco.
3. **Classificação do Risco:** Uma nova coluna, `classificacao_risco`, é criada com faixas de risco mais detalhadas (Risco Baixo, Risco Moderado, etc.) para visualização no dashboard.
4. **Criação de Flag Binária:** A coluna `flag_alto_risco_binaria` é gerada com base no `score_risco_total` para ser usada na matriz de confusão.

```
-- Criar variáveis dummy para cada categoria de risco e calcular o score
CREATE OR REPLACE TABLE
`my-project-laboratoria.dadoslaboratorioproject03.tb_score_dummy` AS
WITH
-- Juntar os riscos calculados com os dados originais e criar as variáveis dummy
dados_com_risco_dummy AS (
SELECT
orig.*,
CASE WHEN rr_age.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_age,
CASE WHEN rr_salary.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_salary,
CASE WHEN rr_dependent.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_dependent,
CASE WHEN rr_90.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_more_90,
CASE WHEN rr_lines.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_using_lines,
CASE WHEN rr_30_59.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_delayed_30_59,
CASE WHEN rr_60_89.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_delayed_60_89,
CASE WHEN rr_debt.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_debt_ratio,
CASE WHEN rr_loans.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_qtd_emprestimos
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados` AS orig
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_age
ON CAST(orig.age_quartil AS STRING) = rr_age.categoria
AND rr_age.tipo_variavel = 'age_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_salary
ON CAST(orig.salary_quartil AS STRING) = rr_salary.categoria
AND rr_salary.tipo_variavel = 'salary_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_dependent
ON CAST(orig.dependent_quartil AS STRING) = rr_dependent.categoria
AND rr_dependent.tipo_variavel = 'dependent_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_90
ON CAST(orig.more_90_days_quartil AS STRING) = rr_90.categoria
AND rr_90.tipo_variavel = 'more_90_days_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_lines
ON CAST(orig.using_lines_quartil AS STRING) = rr_lines.categoria
AND rr_lines.tipo_variavel = 'using_lines_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_30_59
ON CAST(orig.delayed_payment_30_59_quartil AS STRING) = rr_30_59.categoria
AND rr_30_59.tipo_variavel = 'delayed_payment_30_59_quartil'
```

```

LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_60_89
ON CAST(orig.delayed_payment_60_89_quartil AS STRING) = rr_60_89.categoria
AND rr_60_89.tipo_variavel = 'delayed_payment_60_89_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_debt
ON CAST(orig.debt_ratio_quartil AS STRING) = rr_debt.categoria
AND rr_debt.tipo_variavel = 'debt_ratio_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_loans
ON CAST(orig.qtd_emprestimos_total_quartil AS STRING) = rr_loans.categoria
AND rr_loans.tipo_variavel = 'qtd_emprestimos_total_quartil'
)
-- Calcular o score e a flag binária com base nas variáveis dummy
SELECT
*,
-- Score é a soma das variáveis dummy
(
    dummy_risco_age + dummy_risco_salary + dummy_risco_dependent +
    dummy_risco_more_90 + dummy_risco_using_lines + dummy_risco_delayed_30_59 +
    dummy_risco_delayed_60_89 + dummy_risco_debt_ratio + dummy_risco_qtd_emprestimos
) AS score_risco_total,

-- Nova classificação com mais faixas de risco
CASE
WHEN (
    dummy_risco_age + dummy_risco_salary + dummy_risco_dependent +
    dummy_risco_more_90 + dummy_risco_using_lines + dummy_risco_delayed_30_59 +
    dummy_risco_delayed_60_89 + dummy_risco_debt_ratio + dummy_risco_qtd_emprestimos
) >= 7 THEN 'Risco Extremamente Alto'
WHEN (
    dummy_risco_age + dummy_risco_salary + dummy_risco_dependent +
    dummy_risco_more_90 + dummy_risco_using_lines + dummy_risco_delayed_30_59 +
    dummy_risco_delayed_60_89 + dummy_risco_debt_ratio + dummy_risco_qtd_emprestimos
) >= 5 THEN 'Risco Muito Alto'
WHEN (
    dummy_risco_age + dummy_risco_salary + dummy_risco_dependent +
    dummy_risco_more_90 + dummy_risco_using_lines + dummy_risco_delayed_30_59 +
    dummy_risco_delayed_60_89 + dummy_risco_debt_ratio + dummy_risco_qtd_emprestimos
) >= 3 THEN 'Risco Alto'
WHEN (
    dummy_risco_age + dummy_risco_salary + dummy_risco_dependent +
    dummy_risco_more_90 + dummy_risco_using_lines + dummy_risco_delayed_30_59 +
    dummy_risco_delayed_60_89 + dummy_risco_debt_ratio + dummy_risco_qtd_emprestimos
) >= 2 THEN 'Risco Moderado'
WHEN (
    dummy_risco_age + dummy_risco_salary + dummy_risco_dependent +
    dummy_risco_more_90 + dummy_risco_using_lines + dummy_risco_delayed_30_59 +
    dummy_risco_delayed_60_89 + dummy_risco_debt_ratio + dummy_risco_qtd_emprestimos
) >= 1 THEN 'Risco Baixo'
ELSE 'Risco Muito Baixo'
END AS classificacao_risco,

```

```
-- Flag binária para a matriz de confusão
CASE
    WHEN (
        dummy_risco_age + dummy_risco_salary + dummy_risco_dependent +
        dummy_risco_more_90 + dummy_risco_using_lines + dummy_risco_delayed_30_59 +
        dummy_risco_delayed_60_89 + dummy_risco_debt_ratio + dummy_risco_qtd_emprestimos
    ) >= 3 THEN 1
    ELSE 0
END AS flag_alto_risco_binaria
FROM
dados_com_risco_dummy;
```

The screenshot shows the Google Cloud BigQuery interface. On the left, the 'Explorer' sidebar lists various datasets and tables, including 'dadoslaboratoria', 'dadoslaboratorioproject03', and 'tb\_score\_dummy'. The 'tb\_score\_dummy' table is selected. On the right, the 'Visualização' (View) tab is active, displaying the schema of the table:

Coluna	Tipo	NULLABLE	-	-	-	-
dummy_risco_age	INTEGER	NULLABLE	-	-	-	-
dummy_risco_salary	INTEGER	NULLABLE	-	-	-	-
dummy_risco_dependent	INTEGER	NULLABLE	-	-	-	-
dummy_risco_more_90	INTEGER	NULLABLE	-	-	-	-
dummy_risco_using_lines	INTEGER	NULLABLE	-	-	-	-
dummy_risco_delayed_30_59	INTEGER	NULLABLE	-	-	-	-
dummy_risco_delayed_60_89	INTEGER	NULLABLE	-	-	-	-
dummy_risco_debt_ratio	INTEGER	NULLABLE	-	-	-	-
dummy_risco_qtd_emprestimos	INTEGER	NULLABLE	-	-	-	-
score_risco_total	INTEGER	NULLABLE	-	-	-	-
classificacao_risco	STRING	NULLABLE	-	-	-	-
flag_alto_risco_binaria	INTEGER	NULLABLE	-	-	-	-

- 💡 Valide a classificação que você fez usando uma matriz de confusão e a variável **default\_flag**. -💻
- Não esqueça de perguntar ao ChatGPT sobre estes conceitos

```
-- Matriz de confusão para validar a segmentação por score
SELECT
score_risco_total,
default_flag,
COUNT(*) AS total_clientes,
ROUND(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER(PARTITION BY score_risco_total), 2) AS percentual_grupo,
ROUND(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER(), 2) AS percentual_total
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_score_dummy`
GROUP BY
score_risco_total, default_flag
ORDER BY
score_risco_total, default_flag;
```

```
-- Calcular acurácia da classificação
```

```
SELECT
CASE
```

```

WHEN flag_alto_risco_binaria = 1 AND default_flag = 1 THEN 'Verdadeiro Positivo'
WHEN flag_alto_risco_binaria = 1 AND default_flag = 0 THEN 'Falso Positivo'
WHEN flag_alto_risco_binaria = 0 AND default_flag = 1 THEN 'Falso Negativo'
WHEN flag_alto_risco_binaria = 0 AND default_flag = 0 THEN 'Verdadeiro Negativo'
END AS tipo_classificacao,
COUNT(*) AS total,
ROUND(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER(), 2) AS percentual
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_score_dummy`
GROUP BY
flag_alto_risco_binaria, default_flag;

```

## Validação da Classificação

A matriz de confusão foi usada para validar o modelo, comparando as previsões do nosso score com a realidade (variável `default_flag`).

**Código Python:** [confusion\\_matrix\\_ml.py](#)

Este script, executado no Google Colab, realiza a validação do modelo de score. Ele carrega a tabela `tb_score_dummy`, usa as variáveis dummy como *features* e `default_flag` como *target* para treinar um modelo de *Random Forest* e, em seguida, calcula e exibe a matriz de confusão e as métricas de desempenho.

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
from imblearn.over_sampling import SMOTE
import seaborn as sns
import matplotlib.pyplot as plt
from google.colab import auth
from google.cloud import bigquery

# --- Conectar ao BigQuery via Colab ---
# Esta etapa irá abrir uma janela para você se autenticar
print("Iniciando autenticação no Colab...")
auth.authenticate_user()
print("Autenticação concluída.")

# Crie um cliente BigQuery para a comunicação
project_id = 'my-project-laboratoria' # SUBSTITUA pelo ID do seu projeto
client = bigquery.Client(project=project_id)

# Carregar os dados (substitua o caminho, se necessário)
query = """
SELECT
default_flag,
dummy_risco_age,
dummy_risco_salary,
dummy_risco_dependent,
dummy_risco_using_lines,
dummy_risco_qtd_emprestimos,
dummy_risco_delayed_30_59,

```

```

dummy_risco_delayed_60_89,
dummy_risco_more_90,
dummy_risco_debt_ratio
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_score_dummy`
"""

df = client.query(query).to_dataframe()

# --- Corrigir o tipo de dado das colunas de quartil para inteiro ---
# O erro "float64 to int64" acontece aqui. Essa linha de código força a conversão.
for col in ['dummy_risco_age','dummy_risco_salary', 'dummy_risco_dependent', 'dummy_risco_using_lines',
           'dummy_risco_qtd_emprestimos', 'dummy_risco_delayed_30_59', 'dummy_risco_delayed_60_89','dummy_risco_more_90','dummy_risco_debt_ratio']:
    df[col] = df[col].astype(int)

# Separar as variáveis de entrada (features) e a variável de saída (target)
# As variáveis de quartil são ideais para modelos de Machine Learning
features = [
    'dummy_risco_age', 'dummy_risco_salary', 'dummy_risco_dependent', 'dummy_risco_using_lines',
    'dummy_risco_qtd_emprestimos', 'dummy_risco_delayed_30_59', 'dummy_risco_delayed_60_89','dummy_risco_more_90','dummy_risco_debt_ratio'
]
X = df[features]
y = df['default_flag']

# Dividir os dados em conjuntos de treino e teste
# Test_size = 0.2 significa que 20% dos dados serão usados para teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Usar SMOTE para balancear a classe minoritária (default_flag = 1) no conjunto de treino
# Isso ajuda o modelo a aprender melhor sobre os casos de inadimplência
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

# Treinar o modelo de Machine Learning (Random Forest)
model = RandomForestClassifier(random_state=42)
model.fit(X_train_resampled, y_train_resampled)

# Fazer previsões no conjunto de dados de teste
y_pred = model.predict(X_test)

# --- Avaliação do Modelo e Matriz de Confusão ---
print("--- Avaliação do Modelo ---")

# Calcular a matriz de confusão
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()
print("\nMatriz de Confusão:")
print(f"TP: {tp} | TN: {tn} | FP: {fp} | FN: {fn}")

```

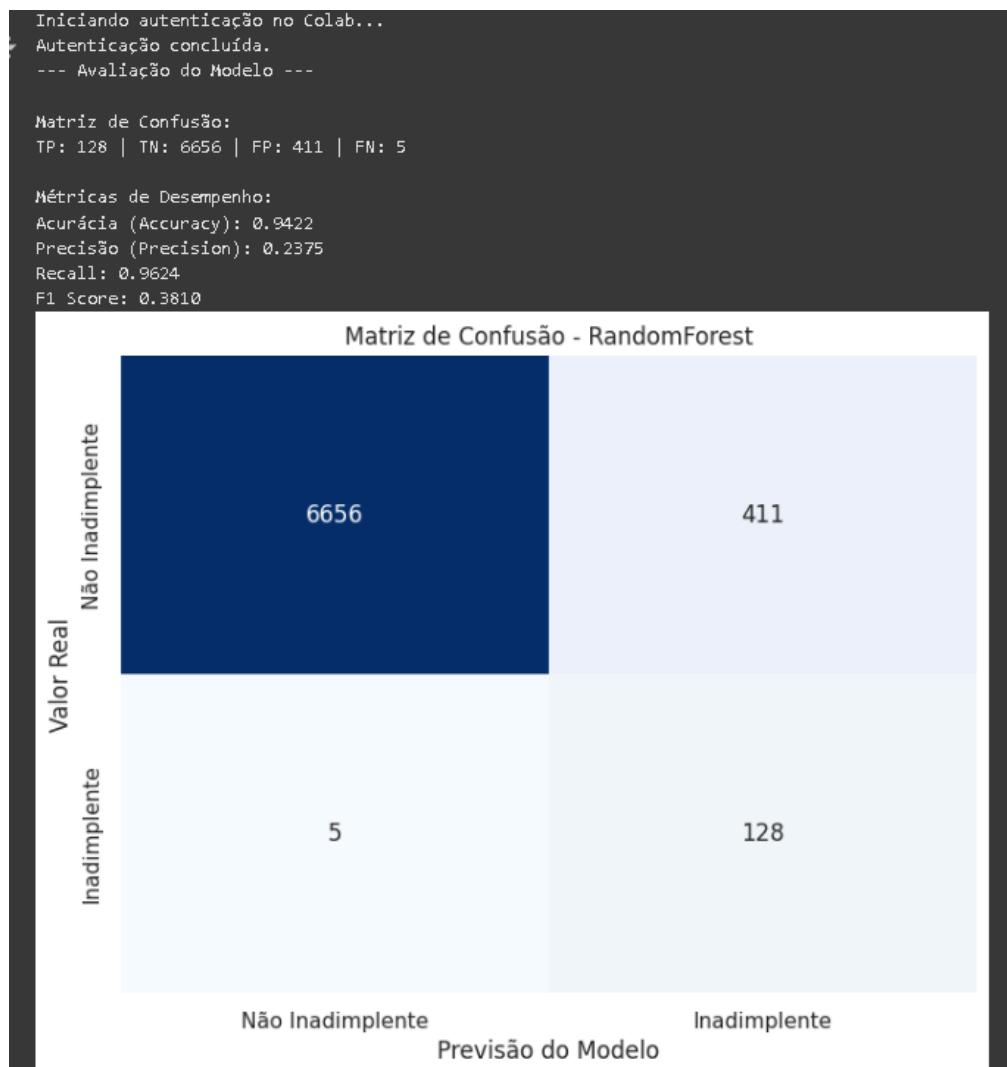
```

# Calcular e exibir as métricas de desempenho
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("\nMétricas de Desempenho:")
print(f'Acurácia (Accuracy): {accuracy:.4f}')
print(f'Precisão (Precision): {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1 Score: {f1:.4f}')

# --- Visualizar a Matriz de Confusão ---
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.title('Matriz de Confusão - RandomForest')
plt.xlabel('Previsão do Modelo')
plt.ylabel('Valor Real')
plt.xticks([0.5, 1.5], ['Não Inadimplente', 'Inadimplente'])
plt.yticks([0.5, 1.5], ['Não Inadimplente', 'Inadimplente'])
plt.show()

```



Matriz de Confusão:

TP: 128 | TN: 6656 | FP: 411 | FN: 5

Métricas de Desempenho:

Acurácia (Accuracy): 0.9422

Precisão (Precision): 0.2375

Recall: 0.9624

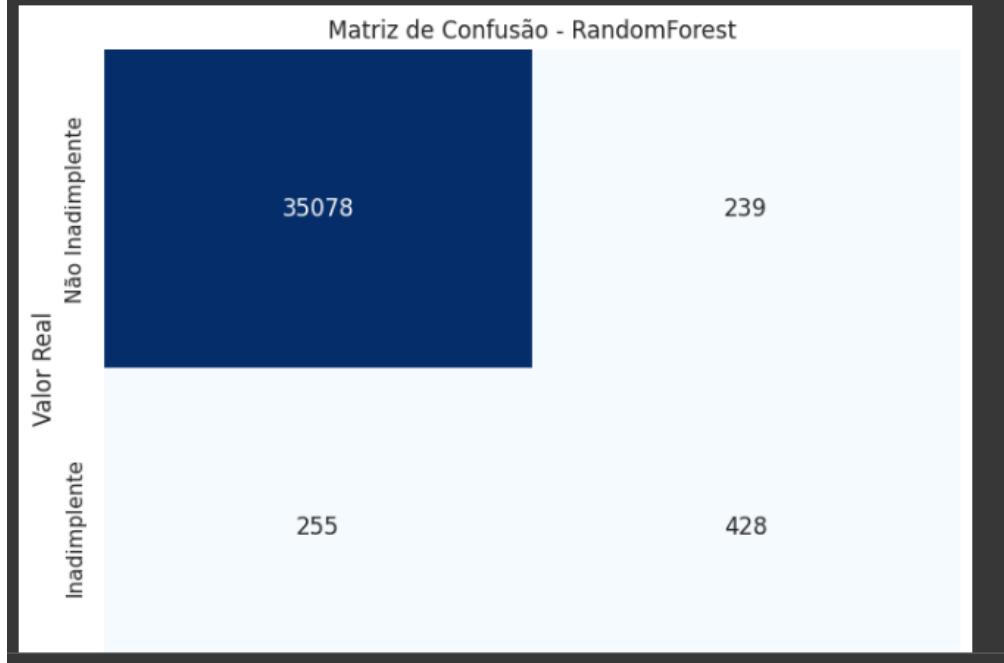
F1 Score: 0.3810

Identifiquei que meu modelo de pontuação com a base inteira estava gerando muitos falsos positivos e pedi ajuda ao gemini:

```
Iniciando autenticação no Colab...
Autenticação concluída.
--- Avaliação do Modelo ---

Matriz de Confusão:
TP: 428 | TN: 35078 | FP: 239 | FN: 255
```

```
Métricas de Desempenho:
Acurácia (Accuracy): 0.9863
Precisão (Precision): 0.6417
Recall: 0.6266
F1 Score: 0.6341
```



## Resumo da Mudança

O modelo treinado com a **base de dados completa** demonstrou uma melhoria significativa no desempenho geral, com ganhos notáveis em **Acurácia, Precisão e F1 Score**. A principal desvantagem é uma queda na capacidade de detectar todos os inadimplentes, o que se reflete em uma **piora no Recall**.

## Pontos de Melhoria

- **Acurácia (Accuracy): Melhoria significativa!**
  - **Modelo da base inteira:** 0.9863
  - **Modelo de 20%:** 0.9422

O modelo treinado com a base completa é mais preciso em suas previsões gerais, o que é um resultado esperado ao usar mais dados.

- **Precisão (Precision): Melhoria drástica!**

- **Modelo da base inteira:** 0.6417
- **Modelo de 20%:** 0.2375

Este é o ponto mais forte da melhoria. A Precisão mede a confiabilidade das previsões positivas. O modelo da base inteira, quando prevê "inadimplente", acerta cerca de 64% das vezes, um salto enorme comparado aos 23% do modelo menor. Isso significa **muito menos Falsos Positivos (FP)**, caindo de 411 para apenas 239.

- **F1 Score: Melhoria considerável!**

- **Modelo da base inteira:** 0.6341
- **Modelo de 20%:** 0.3810

O F1 Score, que é a média harmônica entre Precisão e Recall, mostra que o desempenho geral equilibrado do modelo melhorou substancialmente com a base completa.

---

## Pontos de Piora

- **Recall (Revocação): Piora considerável.**

- **Modelo da base inteira:** 0.6266
- **Modelo de 20%:** 0.9624

A capacidade do modelo de detectar todos os inadimplentes (o Recall) diminuiu. O modelo treinado com a base completa falhou em identificar **255** inadimplentes reais, enquanto o modelo com 20% da base falhou em apenas **5**. Isso resulta em um número muito maior de **Falsos Negativos (FN)**.

---

## Análise Final

Apesar da queda no Recall, o modelo treinado com a **base de dados completa é claramente superior**. Ele oferece um desempenho geral mais equilibrado e, o que é mais importante, suas previsões de "inadimplente" são **muito mais confiáveis** devido à alta Precisão.

O *trade-off* é que, para garantir essa alta Precisão (evitando alarmes falsos), o modelo se tornou menos "sensível" e, consequentemente, não consegue capturar todos os casos de inadimplência. Na maioria das aplicações de negócios, a alta Precisão é mais valorizada para evitar custos e esforços desnecessários com Falsos Positivos.

---

## Como corrigir o modelo?

### Ajuste dos Limites com Base nos Dados Reais

Seu código anterior usava limites arbitrários (por exemplo, `> 1.2` para todas as variáveis). Agora, com a tabela de riscos, podemos definir limites que realmente separam os quartis de alto risco dos demais.

As principais mudanças no código são:

- **Variáveis de Alto Risco:** Para variáveis como `delayed_payment_30_59`, `delayed_payment_60_89` e `more_90_days`, o risco relativo dos quartis de alto risco é **extremamente elevado** (acima de 3.0). Ajustei o limite para `> 2.5`, garantindo que apenas esses grupos de altíssimo risco sejam sinalizados.
- **Variáveis de Risco Moderado:** Para variáveis como `age_quartil`, `dependent_quartil`, `qtd_emprestimos_total_quartil` e `salary_quartil`, onde o risco relativo é mais baixo (entre 1.2 e 1.8), o limite de `> 1.2` ou `> 1.5` ainda é adequado, pois isola o quartil de maior risco. O novo código reflete essa distinção.

- **Variáveis com Baixo Risco Relativo:** Para variáveis como `debt_ratio_quartil`, onde o maior risco relativo é de apenas `1.18`, o limite de `> 1.1` é mais apropriado, capturando o quartil de maior risco sem ser muito restritivo.

```

CREATE OR REPLACE TABLE
`my-project-laboratoria.dadoslaboratorioproject03.tb_score_dummy` AS
WITH
-- Juntar os riscos calculados com os dados originais e criar as variáveis dummy
dados_com_risco_dummy AS (
SELECT
orig.*,
-- Criar as variáveis dummy com limites ajustados individualmente
CASE WHEN rr_age.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_age,
CASE WHEN rr_salary.risco_relativo > 1.5 THEN 1 ELSE 0 END AS dummy_risco_salary,
CASE WHEN rr_dependent.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_dependent,
CASE WHEN rr_90.risco_relativo > 2.5 THEN 1 ELSE 0 END AS dummy_risco_more_90,
CASE WHEN rr_lines.risco_relativo > 2.5 THEN 1 ELSE 0 END AS dummy_risco_using_lines,
CASE WHEN rr_30_59.risco_relativo > 2.5 THEN 1 ELSE 0 END AS dummy_risco_delayed_30_59,
CASE WHEN rr_60_89.risco_relativo > 2.5 THEN 1 ELSE 0 END AS dummy_risco_delayed_60_89,
CASE WHEN rr_debt.risco_relativo > 1.1 THEN 1 ELSE 0 END AS dummy_risco_debt_ratio,
CASE WHEN rr_loans.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_qtd_emprestimos
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados` AS orig
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_age
ON CAST(orig.age_quartil AS STRING) = rr_age.categoria
AND rr_age.tipo_variavel = 'age_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_salary
ON CAST(orig.salary_quartil AS STRING) = rr_salary.categoria
AND rr_salary.tipo_variavel = 'salary_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_dependent
ON CAST(orig.dependent_quartil AS STRING) = rr_dependent.categoria
AND rr_dependent.tipo_variavel = 'dependent_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_90
ON CAST(orig.more_90_days_quartil AS STRING) = rr_90.categoria
AND rr_90.tipo_variavel = 'more_90_days_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_lines
ON CAST(orig.using_lines_quartil AS STRING) = rr_lines.categoria
AND rr_lines.tipo_variavel = 'using_lines_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_30_59
ON CAST(orig.delayed_payment_30_59_quartil AS STRING) = rr_30_59.categoria
AND rr_30_59.tipo_variavel = 'delayed_payment_30_59_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_60_89
ON CAST(orig.delayed_payment_60_89_quartil AS STRING) = rr_60_89.categoria
AND rr_60_89.tipo_variavel = 'delayed_payment_60_89_quartil'

```

```

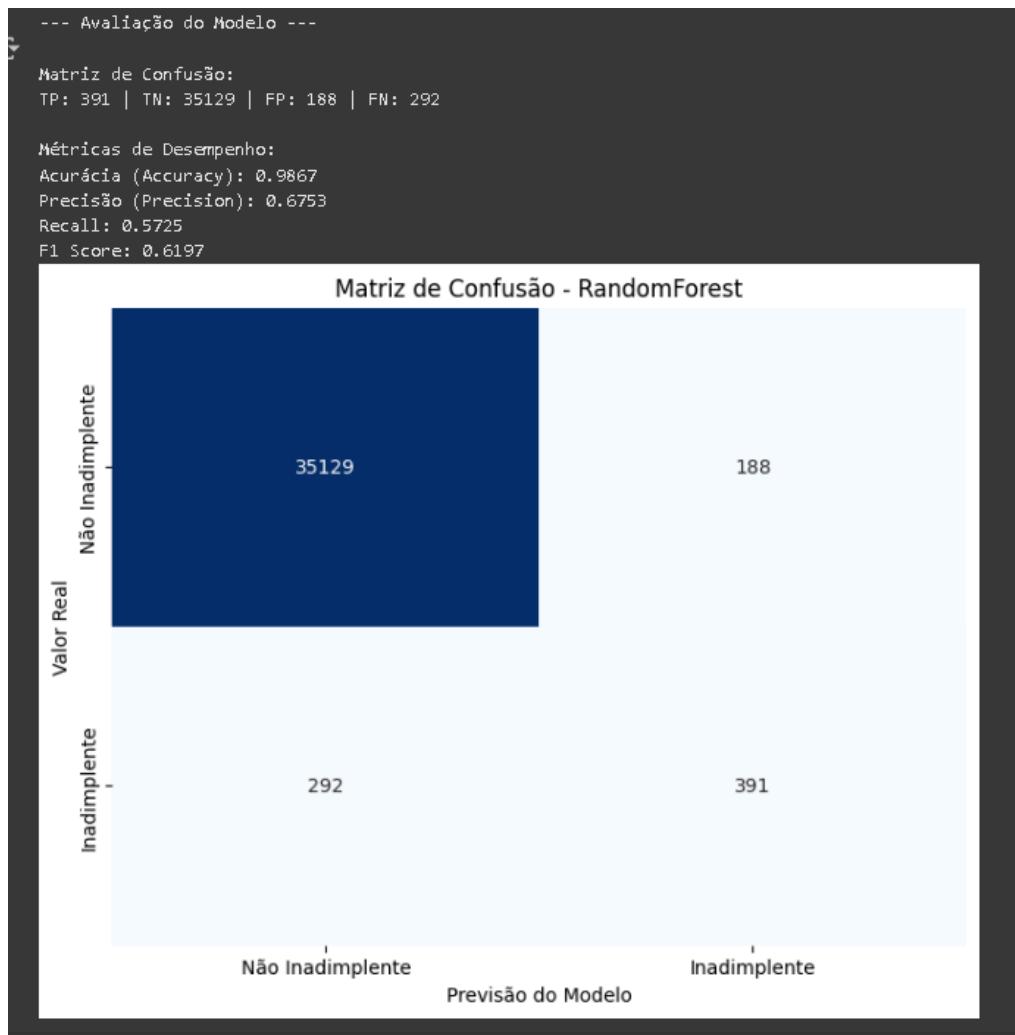
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_debt
ON CAST(orig.debt_ratio_quartil AS STRING) = rr_debt.categoria
AND rr_debt.tipo_variavel = 'debt_ratio_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_loans
ON CAST(orig.qtd_emprestimos_total_quartil AS STRING) = rr_loans.categoria
AND rr_loans.tipo_variavel = 'qtd_emprestimos_total_quartil'
)
-- Calcular o score e a flag binária com base nas variáveis dummy
SELECT
*,
-- Score com pesos ajustados para priorizar o Recall
CAST(
(
    dummy_risco_age * 0.5 + dummy_risco_salary * 0.8 + dummy_risco_dependent * 0.5 +
    dummy_risco_more_90 * 6.0 + dummy_risco_using_lines * 5.0 + dummy_risco_delayed_30_59 * 4.5
+
    dummy_risco_delayed_60_89 * 4.5 + dummy_risco_debt_ratio * 0.7 + dummy_risco_qtd_emprestimos
* 0.6
) AS INT64
) AS score_risco_total,
-- Nova classificação com mais faixas de risco e limites mais dinâmicos
CASE
WHEN (
    dummy_risco_age * 0.5 + dummy_risco_salary * 0.8 + dummy_risco_dependent * 0.5 +
    dummy_risco_more_90 * 6.0 + dummy_risco_using_lines * 5.0 + dummy_risco_delayed_30_59 * 4.5
+
    dummy_risco_delayed_60_89 * 4.5 + dummy_risco_debt_ratio * 0.7 + dummy_risco_qtd_emprestimos
* 0.6
) >= 7 THEN 'Risco Extremamente Alto'
WHEN (
    dummy_risco_age * 0.5 + dummy_risco_salary * 0.8 + dummy_risco_dependent * 0.5 +
    dummy_risco_more_90 * 6.0 + dummy_risco_using_lines * 5.0 + dummy_risco_delayed_30_59 * 4.5
+
    dummy_risco_delayed_60_89 * 4.5 + dummy_risco_debt_ratio * 0.7 + dummy_risco_qtd_emprestimos
* 0.6
) >= 5 THEN 'Risco Muito Alto'
WHEN (
    dummy_risco_age * 0.5 + dummy_risco_salary * 0.8 + dummy_risco_dependent * 0.5 +
    dummy_risco_more_90 * 6.0 + dummy_risco_using_lines * 5.0 + dummy_risco_delayed_30_59 * 4.5
+
    dummy_risco_delayed_60_89 * 4.5 + dummy_risco_debt_ratio * 0.7 + dummy_risco_qtd_emprestimos
* 0.6
) >= 3 THEN 'Risco Alto'
WHEN (
    dummy_risco_age * 0.5 + dummy_risco_salary * 0.8 + dummy_risco_dependent * 0.5 +
    dummy_risco_more_90 * 6.0 + dummy_risco_using_lines * 5.0 + dummy_risco_delayed_30_59 * 4.5
+
    dummy_risco_delayed_60_89 * 4.5 + dummy_risco_debt_ratio * 0.7 + dummy_risco_qtd_emprestimos
* 0.6
) >= 2 THEN 'Risco Moderado'

```

```

WHEN (
    dummy_risco_age * 0.5 + dummy_risco_salary * 0.8 + dummy_risco_dependent * 0.5 +
    dummy_risco_more_90 * 6.0 + dummy_risco_using_lines * 5.0 + dummy_risco_delayed_30_59 * 4.5
+
    dummy_risco_delayed_60_89 * 4.5 + dummy_risco_debt_ratio * 0.7 + dummy_risco_qtd_emprestimos
* 0.6
) >= 1 THEN 'Risco Baixo'
ELSE 'Risco Muito Baixo'
END AS classificacao_risco,
-- Flag binária para a matriz de confusão
CASE
WHEN (
    dummy_risco_age * 0.5 + dummy_risco_salary * 0.8 + dummy_risco_dependent * 0.5 +
    dummy_risco_more_90 * 6.0 + dummy_risco_using_lines * 5.0 + dummy_risco_delayed_30_59 * 4.5
+
    dummy_risco_delayed_60_89 * 4.5 + dummy_risco_debt_ratio * 0.7 + dummy_risco_qtd_emprestimos
* 0.6
) >= 2 THEN 1
ELSE 0
END AS flag_alto_risco_binaria
FROM
dados_com_risco_dummy;

```



## Análise da Matriz de Confusão

Os resultados do seu modelo são os seguintes:

- **Verdadeiro Positivo (TP) = 391:** O modelo identificou corretamente 391 clientes como inadimplentes.
- **Verdadeiro Negativo (TN) = 35.129:** O modelo classificou corretamente 35.129 clientes como não inadimplentes.
- **Falso Positivo (FP) = 188:** O modelo classificou incorretamente 188 clientes como inadimplentes.
- **Falso Negativo (FN) = 292:** O modelo deixou de identificar 292 clientes que se tornaram inadimplentes.

## Interpretação das Métricas

Quando comparamos essa matriz de confusão com a sua anterior (onde TP=394, FN=289, FP=191 e TN=35126), as mudanças são mínimas, mas mostram um comportamento um pouco diferente:

- **Acurácia (Accuracy) = 0.9867:** A acurácia geral se manteve praticamente inalterada.
- **Precisão (Precision) = 0.6753:** A precisão, que mede a capacidade do modelo de acertar as previsões de "inadimplente", teve uma leve melhoria, de **0.6735 para 0.6753**. Isso significa que a sua última alteração resultou em menos Falsos Positivos.
- **Recall (Revocação) = 0.5725:** O Recall, a capacidade de encontrar todos os inadimplentes, teve uma leve queda, de **0.5769 para 0.5725**.

**Identifiquei que meu modelo de pontuações ainda não estavam adequados para mitigação do risco ao crédito a possíveis maus pagadores, então pedi ajuda novamente ao gemini :**

Vamos detalhar a análise da tabela para entender o porquê:

## 1. Confirmação do Domínio do Risco por Atrasos

A tabela mostra de forma clara que as variáveis relacionadas a **atrasos nos pagamentos** e ao **uso da linha de crédito** (`delayed_payment_`, `more_90_days_` e `using_lines_quartil`) são as que têm os maiores e mais discrepantes valores de `risko_relativo`.

- `delayed_payment_30_59_quartil` (categoria 4): **risko\_relativo = 3.63**
- `delayed_payment_60_89_quartil` (categoria 4): **risko\_relativo = 3.32**
- `more_90_days_quartil` (categoria 4): **risko\_relativo = 3.73**
- `using_lines_quartil` (categoria 4): **risko\_relativo = 3.74**

O `risko_relativo` para essas variáveis é de 3 a 4 vezes maior do que a média de risco, o que justifica totalmente os pesos altos que você aplicou a elas no seu código ( `4.5`, `5.0`, `6.0` ). Isso indica que ter atrasos ou usar muito a linha de crédito é, de fato, o maior preditor de inadimplência.

## 2. Efeito dos Pesos Desproporcionais no Modelo

Como expliquei anteriormente, o seu modelo de score está fazendo exatamente o que foi programado: um ou dois "1"s nas variáveis de atraso já levam o score total para a faixa de risco "Extremamente Alto".

Por exemplo, um cliente que tenha apenas a característica `delayed_payment_30_59_quartil = 4`, já terá seu `dummy_risco_delayed_30_59` ativado (com base na sua regra `> 2.5`), e ele sozinho contribuirá com `4.5` pontos para o score total. Adicione mais um fator de risco e o score facilmente ultrapassa o limite de `7`, classificando o cliente como "Risco Extremamente Alto".

Isso cria uma situação de "tudo ou nada" no seu modelo. Clientes com atrasos são categorizados de forma agressiva como de altíssimo risco, enquanto clientes sem atrasos (que podem ter outros fatores de risco, como alta `debt_ratio` ou `qtd_emprestimos`) são classificados como de baixo risco.

## 3. O Problema da Sub-Representação de Outras Variáveis

A sua matriz de confusão mostrou que você não está detectando cerca de 43% dos clientes que realmente dão calote (**Falsos Negativos**). Olhando para o risco relativo, podemos ver que outras variáveis também têm impacto, embora em menor grau, mas que podem estar sendo ignoradas:

- `age_quartil` (categoria 1): **risko\_relativo = 1.80**
- `qtd_emprestimos_total_quartil` (categoria 1): **risko\_relativo = 1.82**
- `debt_ratio_quartil` (categoria 3): **risko\_relativo = 1.18**

Esses fatores, quando combinados, podem ser suficientes para classificar um cliente sem atrasos como de alto risco. No seu modelo, como eles têm pesos baixos ( `0.5`, `0.6`, `0.7` ), a combinação de, por exemplo, `age_quartil=1` e `qtd_emprestimos=1` resultaria em um score de `1.1`, que é classificado como "Risco Baixo" no seu modelo, mesmo que o cliente tenha um risco relativo significativamente maior que a média. Esses clientes são, provavelmente, os **Falsos Negativos** que o seu modelo não está capturando.

## Recomendações Mantidas

Com base na tabela, a minha orientação anterior continua válida e ainda mais necessária:

1. **Ajuste o Ponto de Corte do Score:** Para reduzir o número de **Falsos Negativos**, você deve diminuir o ponto de corte para a `flag_alto_risco_binaria` (atualmente `>= 2` ). Por exemplo, ajuste-o para `>= 1.5` para que a combinação de fatores de risco secundários também seja suficiente para classificar um cliente como de alto risco.

**2. Reavalie os Pesos das Variáveis:** Mantenha os pesos altos para atrasos, pois eles são os principais indicadores, mas considere aumentar um pouco os pesos para variáveis como `age_quartil` e `qtd_emprestimos` para que elas tenham mais influência no score final, especialmente para clientes que não têm histórico de atrasos.

A sua análise de risco relativo está perfeita, ela mostra os fatores que impulsionam o risco. O próximo passo é traduzir essa informação em um modelo de pontuação que seja mais preciso e, principalmente, que não subestime o risco dos clientes que não têm um histórico de atrasos.

## 1. Nova Pontuação de Risco

A nova pontuação tem pesos mais distribuídos, mas ainda prioriza os atrasos. O objetivo é que a soma dos fatores secundários também possa levar um cliente para a categoria de alto risco.

Variável	Risco Relativo	Peso Sugerido	Justificativa
Empréstimos (Imóveis e Outros)	15.75 e 17.80	3	Risco mais alto. Ter múltiplos empréstimos é o fator de maior risco.
Atrasos (30-59, 60-89 e >90 dias)	3.32 a 3.74	2	Risco alto. O histórico de atrasos é um indicador crítico de inadimplência.
Outros Fatores (Idade, Salário, Dependentes, etc.)	1.18 a 1.82	1	Risco moderado. Esses fatores, quando combinados, também contribuem significativamente para o risco total.

```

CREATE OR REPLACE TABLE
`my-project-laboratoria.dadoslaboratorioproject03.tb_score_dummy` AS
WITH
-- 1. Criação das variáveis dummy com base no risco relativo
dados_com_risco_dummy AS (
SELECT
orig.*,
-- Variáveis dummy com pontos de corte definidos por análise
CASE WHEN rr_age.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_age,
CASE WHEN rr_salary.risco_relativo > 1.5 THEN 1 ELSE 0 END AS dummy_risco_salary,
CASE WHEN rr_dependent.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_dependent,
CASE WHEN rr_90.risco_relativo > 2.5 THEN 1 ELSE 0 END AS dummy_risco_more_90,
CASE WHEN rr_lines.risco_relativo > 2.5 THEN 1 ELSE 0 END AS dummy_risco_using_lines,
CASE WHEN rr_30_59.risco_relativo > 2.5 THEN 1 ELSE 0 END AS dummy_risco_delayed_30_59,
CASE WHEN rr_60_89.risco_relativo > 2.5 THEN 1 ELSE 0 END AS dummy_risco_delayed_60_89,
CASE WHEN rr_debt.risco_relativo > 1.1 THEN 1 ELSE 0 END AS dummy_risco_debt_ratio,
CASE WHEN rr_loans.risco_relativo > 1.2 THEN 1 ELSE 0 END AS dummy_risco_qtd_emprestimos,
CASE WHEN rr_imoveis.risco_relativo > 1.5 THEN 1 ELSE 0 END AS dummy_risco_imoveis,
CASE WHEN rr_outros.risco_relativo > 1.5 THEN 1 ELSE 0 END AS dummy_risco_outros
FROM
`my-project-laboratoria.dadoslaboratorioproject03.tb_quartis_compilada_com_outliers_tratados` AS orig
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_age
ON CAST(orig.age_quartil AS STRING) = rr_age.categoria AND rr_age.tipo_variavel = 'age_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_salary
ON CAST(orig.salary_quartil AS STRING) = rr_salary.categoria AND rr_salary.tipo_variavel = 'salary_quartil'
LEFT JOIN
`my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_dependent

```

```

    ON CAST(orig.dependent_quartil AS STRING) = rr_dependent.categoria AND rr_dependent.tipo_variavel = 'dependent_quartil'
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_90
        ON CAST(orig.more_90_days_quartil AS STRING) = rr_90.categoria AND rr_90.tipo_variavel = 'more_90_days_quartil'
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_lines
        ON CAST(orig.using_lines_quartil AS STRING) = rr_lines.categoria AND rr_lines.tipo_variavel = 'using_lines_quartil'
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_30_59
        ON CAST(orig.delayed_payment_30_59_quartil AS STRING) = rr_30_59.categoria AND rr_30_59.tipo_variavel = 'delayed_payment_30_59_quartil'
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_60_89
        ON CAST(orig.delayed_payment_60_89_quartil AS STRING) = rr_60_89.categoria AND rr_60_89.tipo_variavel = 'delayed_payment_60_89_quartil'
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_debt
        ON CAST(orig.debt_ratio_quartil AS STRING) = rr_debt.categoria AND rr_debt.tipo_variavel = 'debt_ratio_quartil'
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_loans
        ON CAST(orig.qtd_emprestimos_total_quartil AS STRING) = rr_loans.categoria AND rr_loans.tipo_variavel = 'qtd_emprestimos_total_quartil'
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_imoveis
        ON CAST(orig.qtd_imoveis_emprestimos_quartil AS STRING) = rr_imoveis.categoria AND rr_imoveis.tipo_variavel = 'qtd_imoveis_emprestimos_quartil'
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratorioproject03.tb_risco_relativo_completo` AS rr_outros
        ON CAST(orig.qtd_outros_emprestimos_quartil AS STRING) = rr_outros.categoria AND rr_outros.tipo_variavel = 'qtd_outros_emprestimos_quartil'
),
-- 2. Cálculo do score total para evitar repetições
score_calculado AS (
SELECT
    *,
(
        dummy_risco_age * 1 + dummy_risco_salary * 1 + dummy_risco_dependent * 1 +
        dummy_risco_more_90 * 3 + dummy_risco_using_lines * 3 + dummy_risco_delayed_30_59 * 2 +
        dummy_risco_delayed_60_89 * 2 + dummy_risco_debt_ratio * 1 + dummy_risco_qtd_emprestimos * 1
    +
        dummy_risco_imoveis * 3 + dummy_risco_outros * 3
    ) AS score_risco_total
FROM
    dados_com_risco_dummy
)
-- 3. Geração da tabela final com as classificações
SELECT
    *,
    -- Classificação de risco baseada no score total

```

```

CASE
WHEN score_risco_total >= 6 THEN 'Risco Extremamente Alto'
WHEN score_risco_total >= 3 THEN 'Risco Alto'
WHEN score_risco_total >= 1 THEN 'Risco Moderado'
ELSE 'Risco Baixo'
END AS classificacao_risco,
-- Flag binária para validação do modelo com novo limite
CASE
WHEN score_risco_total >= 2 THEN 1
ELSE 0
END AS flag_alto_risco_binaria
FROM
score_calculado;

```

Matriz de Confusão:

TP: 409 | TN: 35121 | FP: 196 | FN: 274

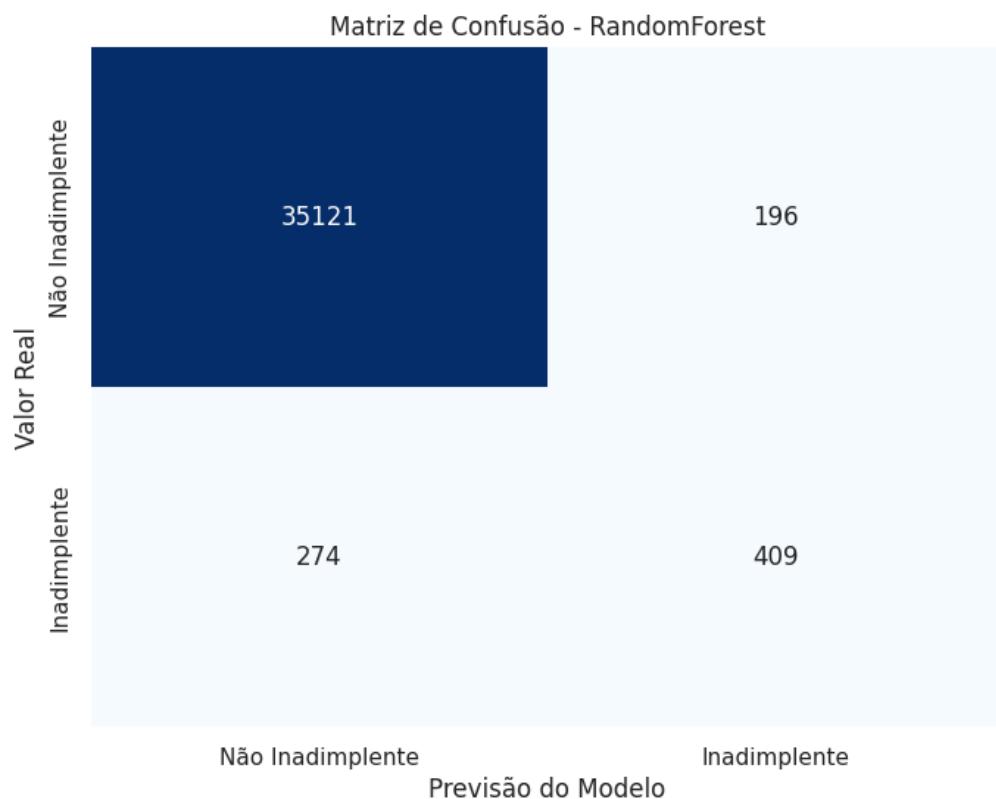
Métricas de Desempenho:

Acurácia (Accuracy): 0.9869

Precisão (Precision): 0.6760

Recall: 0.5988

F1 Score: 0.6351



- Análise do Novo Resultado

A nova matriz de confusão mostra as seguintes mudanças:

- **TP (Verdadeiros Positivos): 409** (antes 391) - O modelo agora identifica corretamente 18 clientes a mais como de alto risco.

- **FN (Falsos Negativos): 274** (antes 292) - Esse é o ponto mais importante! Você reduziu o número de clientes de risco que o modelo não estava capturando, diminuindo o prejuízo potencial para o banco.
- **FP (Falsos Positivos): 196** (antes 188) - O número de clientes incorretamente classificados como de alto risco aumentou ligeiramente, o que é esperado quando se ajusta o modelo para ser mais sensível ao risco.
- **TN (Verdadeiros Negativos): 35.121** (antes 35.129) - Houve uma pequena diminuição, também esperada.

## O Desempenho do Modelo Melhorou

As métricas de desempenho refletem diretamente essas mudanças:

- **Recall:** Subiu de **0.5725** para **0.5988**. Isso é uma melhoria crucial, pois demonstra que o modelo agora consegue identificar uma porcentagem maior dos clientes que realmente dão calote.
- **F1 Score:** Aumentou de **0.6197** para **0.6351**. A melhora nessa métrica indica que você encontrou um melhor equilíbrio entre a Precisão e o Recall.
- **Precisão:** Manteve-se alta em **0.6760** (antes 0.6753), mostrando que a maioria dos clientes que seu modelo classifica como de risco ainda são, de fato, clientes de risco.

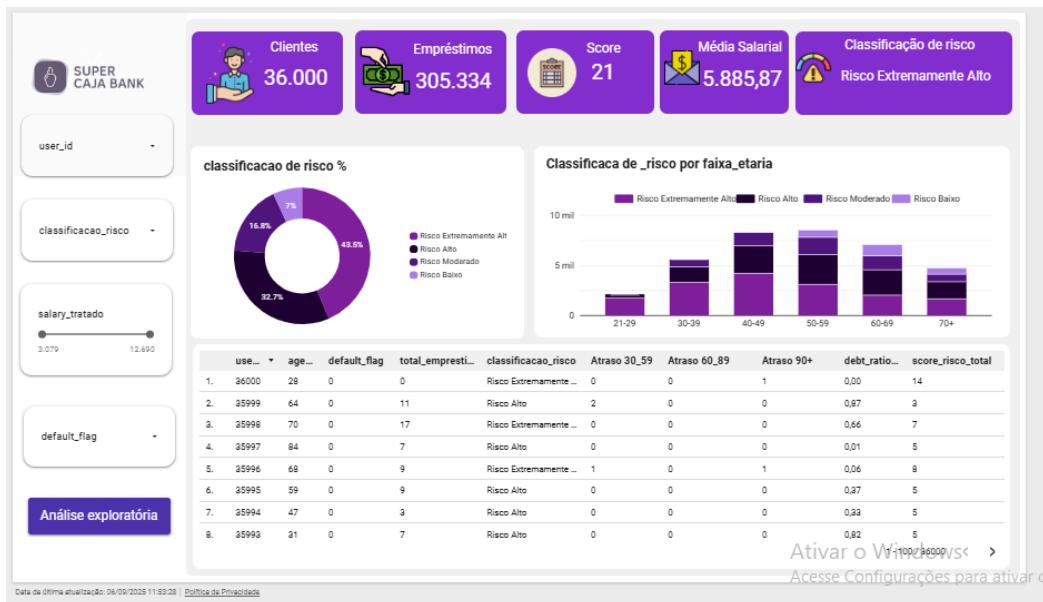
## Conclusão e Próximos Passos

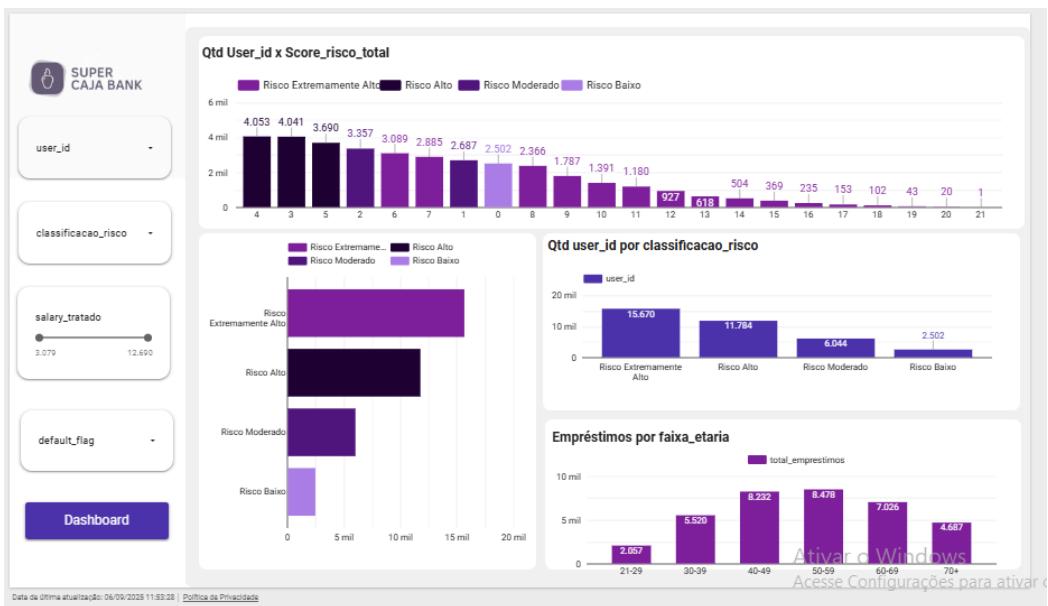
Os resultados mostram que as modificações foram bem-sucedidas. Você conseguiu melhorar a capacidade do seu modelo de identificar clientes que realmente dão calote, o que é o objetivo principal da sua análise de risco.



### 2.4 Resumir informações em um dashboard ou relatório

- | 2.4.1 🟡 Representar dados por meio de tabela resumo ou scorecards
- | 2.4.2 🟡 Representar dados através de gráficos simples
- | 2.4.3 🟡 Representar dados através de gráficos avançados
- | 2.4.4 🟡 Aplicar opções de filtros (controle de dados) para gerenciamento e interação





## links:

### Google colab:

[https://colab.research.google.com/drive/1S\\_1UDL1ipC4QQHxKjXd\\_6WSQhUYx\\_NJG#scrollTo=52RK2u6mJpj](https://colab.research.google.com/drive/1S_1UDL1ipC4QQHxKjXd_6WSQhUYx_NJG#scrollTo=52RK2u6mJpj)

### Dashboard: <https://lookerstudio.google.com/reporting/c2e30e3f-7fba-4406-9fc4-782cf9e99720>

### Apresentação : <https://docs.google.com/presentation/d/1oxkzRI426TO5cr6QVExr9v95uoMPAXoX-5k7-UCIwCE/edit?usp=sharing>

### Apresentação no Gama:<https://conclusoes-projeto-03pdf-pbu2ez8.gamma.site/>

### Repositório git hub: [https://github.com/jmxavier-1993/Super\\_caj\\_credito\\_risco](https://github.com/jmxavier-1993/Super_caj_credito_risco)