Identificar nulos

SELECT

COUNTIF( user\_id IS NULL) AS nulos\_user,

COUNTIF(default\_flag IS NULL) AS nulos\_default

FROM

projeto3-469700.dataset\_banco.default

SELECT

COUNTIF( user\_id IS NULL) AS nulos\_user,

COUNTIF(more\_90\_days\_overdue IS NULL) AS nulos\_days,

COUNTIF(using\_lines\_not\_secured\_personal\_assets IS NULL) AS nulos\_assets,

COUNTIF( number\_times\_delayed\_payment\_loan\_30\_59\_days IS NULL) AS nulos\_30\_59,

COUNTIF( number\_times\_delayed\_payment\_loan\_60\_89\_days IS NULL) AS nulos\_60\_89,

COUNTIF( debt\_ratio IS NULL) AS nulos\_debt

FROM `projeto3-469700.dataset\_banco.loans\_detail`

SELECT

COUNTIF( user\_id IS NULL) AS nulos\_id,

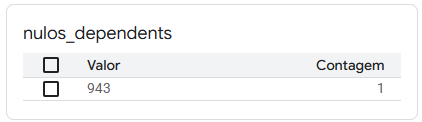
COUNTIF( age IS NULL) AS nulos\_age,

COUNTIF( sex IS NULL) AS nulos\_sex,

COUNTIF( last\_month\_salary IS NULL) AS nulos\_salary,

COUNTIF( number\_dependents IS NULL) AS nulos\_dependents

FROM `projeto3-469700.dataset\_banco.user\_info`



SELECT

COUNTIF( loan\_id IS NULL) AS nulos\_loan,

COUNTIF( user\_id IS NULL) AS nulos\_user,

COUNTIF( loan\_type IS NULL) AS nulos\_type,

FROM `projeto3-469700.dataset\_banco.loans\_outstanding`

remoção de nulos

SELECT \*

FROM `projeto3-469700.dataset\_banco.user\_info`

WHERE user\_id IS NOT NULL

AND age IS NOT NULL

AND sex IS NOT NULL

AND last\_month\_salary IS NOT NULL

AND number\_dependents IS NOT NULL;

SELECT \* EXCEPT (sex),

FROM `projeto3-469700.dataset\_banco.notnull\_info`

identificar duplicados

SELECT user\_id,

COUNT(\*) AS total

FROM `projeto3-469700.dataset\_banco.user\_info`

GROUP BY user\_id

HAVING COUNT(\*) > 1;

SELECT user\_id,

COUNT(\*) AS total

FROM `projeto3-469700.dataset\_banco.default`

GROUP BY user\_id

HAVING COUNT(\*) > 1;

SELECT user\_id,

COUNT(\*) AS total

FROM `projeto3-469700.dataset\_banco.loans\_detail`

GROUP BY user\_id

HAVING COUNT(\*) > 1;

SELECT user\_id,loan\_id,

COUNT(\*) AS total,

FROM `projeto3-469700.dataset\_banco.loans\_outstanding`

GROUP BY user\_id,loan\_id

HAVING COUNT(\*) > 1

Nenhuma das consultas encontrou valores duplicados

correlação

SELECT

CORR(more\_90\_days\_overdue, using\_lines\_not\_secured\_personal\_assets) AS corr\_90\_lines,

CORR(more\_90\_days\_overdue,number\_times\_delayed\_payment\_loan\_30\_59\_days) AS corr\_90\_30,

CORR(more\_90\_days\_overdue,number\_times\_delayed\_payment\_loan\_60\_89\_days) AS corr\_90\_60,

CORR(more\_90\_days\_overdue,debt\_ratio) AS corr\_90\_debt,

CORR(debt\_ratio,number\_times\_delayed\_payment\_loan\_30\_59\_days) AS corr\_debt\_30,

CORR(debt\_ratio,number\_times\_delayed\_payment\_loan\_60\_89\_days) AS corr\_debt\_60,

CORR (debt\_ratio,using\_lines\_not\_secured\_personal\_assets) AS corr\_debt\_lines,

CORR(number\_times\_delayed\_payment\_loan\_30\_59\_days, number\_times\_delayed\_payment\_loan\_60\_89\_days) AS corr\_30\_60),

CORR (number\_times\_delayed\_payment\_loan\_30\_59\_days,using\_lines\_not\_secured\_personal\_assets)AS corr\_30\_lines),

CORR(number\_times\_delayed\_payment\_loan\_60\_89\_days,using\_lines\_not\_secured\_personal\_assets)AS corr\_60\_lines)

FROM `projeto3-469700.dataset\_banco.loans\_detail`

| corr\_30\_60 | corr\_30\_lines | corr\_60\_lines | corr\_90\_30 | corr\_90\_60 | corr\_90\_debt | corr\_90\_lines | corr\_debt\_30 | corr\_debt\_60 | corr\_debt\_lines |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.98655364549866986 | -0.0012496300975538806 | -0.000772448431582398 | 0.98291680661459857 | 0.99217552634075257 | -0.0082076486652037147 | -0.0013609406054606965 | -0.0052226124100982953 | -0.0074023325612315215 | 0.015012138743289461 |

Optei pela remoção das colunas: number\_times\_delayed\_payment\_loan\_60\_89\_days e number\_times\_delayed\_payment\_loan\_30\_59\_days

SELECT \*

EXCEPT (number\_times\_delayed\_payment\_loan\_30\_59\_days,number\_times\_delayed\_payment\_loan\_60\_89\_days), FROM `projeto3-469700.dataset\_banco.loans\_detail`

unificando consultas

SELECT user\_id, age, last\_month\_salary, number\_dependents

FROM projeto3-469700.dataset\_banco.user\_info

UNION ALL

SELECT user\_id, age, last\_month\_salary, number\_dependents

FROM projeto3-469700.dataset\_banco.user\_info

WHERE user\_id IS NOT NULL

AND age IS NOT NULL

AND last\_month\_salary IS NOT NULL

AND number\_dependents IS NOT NULL

A tabela gerada continuou com os valores nulos, então removi a coluna sexo da view com nulos removidos

SELECT \* EXCEPT (sex),

FROM `projeto3-469700.dataset\_banco.notnull\_info`

padronizar variáveis categoricas

SELECT

loan\_id,user\_id,

UPPER(loan\_type) AS LOAN\_TYPE

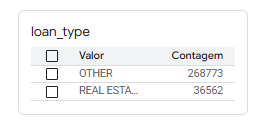
FROM `projeto3-469700.dataset\_banco.loans\_outstanding`

SELECT loan\_id, user\_id,

REPLACE(LOAN\_TYPE, 'OTHERS', 'OTHER') AS loan\_type

FROM

`projeto3-469700.dataset\_banco.UPPER\_loans\_outs`



padronizar variáveis numéricas

-- Etapa 1: Calcular a mediana

WITH mediana AS (

SELECT

APPROX\_QUANTILES(debt\_ratio, 2)[OFFSET(1)] AS med

FROM `projeto3-469700.dataset\_banco.loans\_detail\_limpo`

),

-- Etapa 2: Calcular o desvio absoluto em relação à mediana

diferencas AS (

SELECT

ABS(debt\_ratio - med) AS diff

FROM `projeto3-469700.dataset\_banco.loans\_detail\_limpo`, mediana

),

-- Etapa 3: Calcular o MAD (mediana dos desvios absolutos)

mad\_calc AS (

SELECT

APPROX\_QUANTILES(diff, 2)[OFFSET(1)] AS mad

FROM diferencas

)

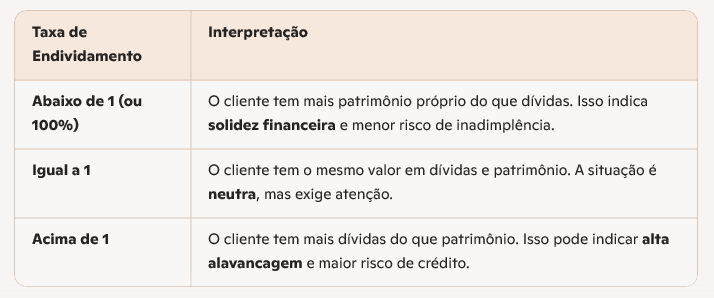
-- Etapa 4: Selecionar os outliers

SELECT \*

FROM `projeto3-469700.dataset\_banco.loans\_detail\_limpo`, mediana, mad\_calc

WHERE ABS(debt\_ratio - med) > 3 \* mad;

O comando encontrou **8231** outliers (taxa de endividamento maior que 1.09)



A mediana calculada para os dados foi de 0,36 e a mediana do desvio absoluto foi 0,24.

Dessa forma, clientes com det\_ratio maior de 1,09 foram incluídos como outliers.

Não considero que seja uma boa estratégia, já que exclui muitos clientes e mantém apenas aqueles com baixíssimo nível de endividamento.

Classifiquei debt\_ratio por quartis e obtive 9000 dados no quartil 4, dentre os dados existiam taxas de débito baixas, já que o quartil apenas divide as taxas em 4 segmentos com a mesma quantidade de dados. Sendo assim, não achei essa uma boa estratégia para eliminar extremos.

Escolhi um valor arbitrário de 10.000 e eliminei os clientes com taxas acima desse valor. Um total de 47 clientes foram eliminados.

SELECT COUNT(\*) AS total\_acima\_dezmil

FROM `projeto3-469700.dataset\_banco.loans\_detail\_limpa`

WHERE debt\_ratio > 10000;



criando novas variáveis

-- Contabilizar a quantidade de empréstimos que cada cliente possui

SELECT

t.\*,

c.total\_loans

FROM

`projeto3-469700.dataset\_banco.loans\_outs\_ok` t

JOIN (

SELECT

user\_id,

COUNT(\*) AS total\_loans

FROM

`projeto3-469700.dataset\_banco.loans\_outs\_ok`

GROUP BY

user\_id

) c

ON t.user\_id = c.user\_id

--- Agrupando as informações por cliente

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.loans\_por\_cliente` AS

SELECT

user\_id,

MAX(total\_loans) AS total\_loans,

ARRAY\_AGG(DISTINCT loan\_type) AS tipos\_emprestimo

FROM

`projeto3-469700.dataset\_banco.loans\_out\_new\_var`

GROUP BY

user\_id

unir tabelas

-- Unindo as quatro tabelas tratadas

CREATE OR REPLACE TABLE projeto3-469700.dataset\_banco.tabela\_unificada AS

SELECT

lpc.\*,

ldr.\* EXCEPT(user\_id),

uil.\* EXCEPT(user\_id),

dft.\* EXCEPT(user\_id)

FROM projeto3-469700.dataset\_banco.loans\_por\_cliente AS lpc

LEFT JOIN projeto3-469700.dataset\_banco.loans\_detail\_replace AS ldr

ON lpc.user\_id = ldr.user\_id

LEFT JOIN projeto3-469700.dataset\_banco.user\_info\_limpa AS uil

ON lpc.user\_id = uil.user\_id

LEFT JOIN projeto3-469700.dataset\_banco.default AS dft

ON lpc.user\_id = dft.user\_id;

Apareceu um monte de valores null em algumas colunas

-- Comando atualizado para unificar a partir da tabela detail\_limpa e evitar os nulos que já haviam sido limpos:

CREATE OR REPLACE TABLE projeto3-469700.dataset\_banco.tabela\_unificada AS

SELECT

uil.\*,

lpc.\* EXCEPT(user\_id),

ldr.\* EXCEPT(user\_id),

dft.\* EXCEPT(user\_id)

FROM projeto3-469700.dataset\_banco.user\_info\_limpa AS uil

LEFT JOIN projeto3-469700.dataset\_banco.loans\_por\_cliente AS lpc

ON uil.user\_id = lpc.user\_id

LEFT JOIN projeto3-469700.dataset\_banco.loans\_detail\_replace AS ldr

ON uil.user\_id = ldr.user\_id

LEFT JOIN projeto3-469700.dataset\_banco.default AS dft

ON uil.user\_id = dft.user\_id;

-- Comando para substituição dos nulos remanescentes na coluna loans\_total:

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.tabela\_unificada` AS

SELECT

user\_id,

age,

last\_month\_salary,

number\_dependents,

debt\_ratio,

quartil\_debt\_ratio,

tipos\_emprestimo,

using\_lines\_not\_secured\_personal\_assets,

default\_flag,

more\_90\_days\_overdue,

COALESCE(total\_loans,0) AS loans\_total

FROM `projeto3-469700.dataset\_banco.tabela\_unificada`

quartil

-- Separando as variáveis por quartis

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.tabela\_QUARTIS` AS

SELECT \*,

NTILE(4) OVER (ORDER BY age) AS quartil\_age,

NTILE(4) OVER (ORDER BY last\_month\_salary) AS quartil\_salary,

NTILE(4) OVER (ORDER BY loans\_total) AS quartil\_loan,

NTILE(4) OVER (ORDER BY using\_lines\_not\_secured\_personal\_assets) AS quartil\_lines,

NTILE(4) OVER (ORDER BY debt\_ratio) AS quartil\_debt,

NTILE(4) OVER (ORDER BY number\_dependents) AS quartil\_dependents,

NTILE(4) OVER (ORDER BY more\_90\_days\_overdue) AS quartil\_days

FROM `projeto3-469700.dataset\_banco.tabela\_unificada`

corr

-- Calculando a correlação para todas as possíveis combinações de variáveis e quartis

SELECT

CORR(quartil\_debt, quartil\_age) AS corr\_qdebt\_qage,

CORR(quartil\_debt, quartil\_salary) AS corr\_qdebt\_qsalary,

CORR(quartil\_debt, quartil\_lines) AS corr\_qdebt\_qlines,

CORR(quartil\_debt, quartil\_loan) AS corr\_qdebt\_qloan,

CORR(quartil\_debt, quartil\_days) AS corr\_qdebt\_qdays,

CORR(quartil\_debt, quartil\_dependents) AS corr\_qdebt\_qdependents,

CORR(quartil\_age, quartil\_salary) AS corr\_qage\_qsalary,

CORR(quartil\_age, quartil\_lines) AS corr\_qage\_qlines,

CORR(quartil\_age, quartil\_loan) AS corr\_qage\_qloan,

CORR(quartil\_age, quartil\_days) AS corr\_qage\_qdays,

CORR(quartil\_age,quartil\_dependents) AS corr\_qage\_qdependents,

CORR(quartil\_salary,quartil\_lines) AS corr\_qsalary\_qlines,

CORR(quartil\_salary,quartil\_loan) AS corr\_qsalary\_qloan,

CORR(quartil\_salary,quartil\_days) AS corr\_qsalary\_qdays,

CORR(quartil\_salary,quartil\_dependents) AS corr\_qsalary\_qdependents,

CORR(quartil\_lines,quartil\_loan) AS corr\_qlines\_qloan,

CORR(quartil\_lines,quartil\_days) AS corr\_qlines\_days,

CORR(quartil\_lines,quartil\_dependents) AS corrqlines\_qdependents,

CORR(quartil\_loan,quartil\_days) AS corr\_qloan\_qdays,

CORR(quartil\_loan,quartil\_dependents) AS corr\_qloan\_qdependents,

CORR(quartil\_days,quartil\_dependents) AS corr\_qdays\_qdependents,

CORR(debt\_ratio, age) AS corr\_debt\_age,

CORR(debt\_ratio, last\_month\_salary) AS corr\_debt\_salary,

CORR(debt\_ratio, more\_90\_days\_overdue) AS corr\_debt\_days,

CORR(debt\_ratio, number\_dependents) AS corr\_debt\_dependents,

CORR(debt\_ratio, loans\_total) AS corr\_debt\_loans,

CORR(debt\_ratio, default\_flag) AS corr\_debt\_flag,

CORR(debt\_ratio, using\_lines\_not\_secured\_personal\_assets) AS corr\_debt\_lines,

CORR(default\_flag, age) AS corr\_flag\_age,

CORR(default\_flag, last\_month\_salary) AS corr\_flag\_salary,

CORR(default\_flag, more\_90\_days\_overdue) AS corr\_flag\_days,

CORR(default\_flag, number\_dependents) AS corr\_flag\_dependents,

CORR(default\_flag, loans\_total) AS corr\_flag\_loans,

CORR(default\_flag, using\_lines\_not\_secured\_personal\_assets) AS corr\_flag\_lines,

CORR(age, last\_month\_salary) AS corr\_age\_flag,

CORR(age, more\_90\_days\_overdue) AS corr\_age\_days,

CORR(age, number\_dependents) AS corr\_age\_dependents,

CORR(age, loans\_total) AS corr\_age\_loans,

CORR(age, using\_lines\_not\_secured\_personal\_assets) AS corr\_age\_lines,

CORR(last\_month\_salary, more\_90\_days\_overdue) AS corr\_salary\_days,

CORR(last\_month\_salary, number\_dependents) AS corr\_salary\_dependents,

CORR(last\_month\_salary, loans\_total) AS corr\_salary\_loans,

CORR(last\_month\_salary, using\_lines\_not\_secured\_personal\_assets) AS corr\_salary\_lines,

CORR (more\_90\_days\_overdue, number\_dependents) AS corr\_days\_dependents,

CORR (more\_90\_days\_overdue,loans\_total) AS corr\_days\_loans,

CORR (more\_90\_days\_overdue, using\_lines\_not\_secured\_personal\_assets) AS corr\_days\_lines,

CORR (number\_dependents, loans\_total) AS corr\_dependents\_loans,

CORR (number\_dependents, using\_lines\_not\_secured\_personal\_assets) AS corr\_dependents\_lines,

CORR(loans\_total, using\_lines\_not\_secured\_personal\_assets) AS corr\_loans\_lines

FROM `projeto3-469700.dataset\_banco.tabela\_QUARTIS`

Calculei a correlação para todas as combinações de variáveis e quartis. Encontrei uma correlação de 0,83 entre os quartis de total de empréstimos e atrasos com mais de 90 dias.

Avaliando mais a fundo, percebi que a decisão de manter apenas a coluna com o maior atraso no pagamento talvez não tenha sido a melhor, porque a grande maioria dos clientes nunca atrasaram mais do que 90 dias. Com isso, mesmo no quartil 4 tenho clientes que nunca tiveram atrasos superiores e essa coluna acaba não sendo uma boa opção para basear meus cálculos.

Diante disso, vou realizar o cálculo de risco relativo para todas as minhas demais variáveis, ainda que não tenha encontrado uma boa correlação entre elas. A partir do risco relativo vou calcular o score dos meus clientes, desconsiderando a variável de atrasos superior a 90 dias.

RR\_quartis

incidência e risco

-- Calculo da taxa de incidência

WITH base AS (

SELECT

quartil\_xx,

COUNT(\*) AS total,

COUNTIF(default\_flag = 1) AS inadimplentes,

SAFE\_DIVIDE(COUNTIF(default\_flag = 1), COUNT(\*)) AS taxa\_inadimplencia

FROM `projeto3-469700.dataset\_banco.tabela\_QUARTIS`

GROUP BY quartil\_xx

),

combinado AS (

SELECT

a.quartil\_xx AS quartil\_base,

-- Dados do quartil avaliado

a.inadimplentes AS inadimplentes\_base,

a.total AS total\_base,

a.taxa\_inadimplencia AS taxa\_base,

-- Dados combinados dos demais quartis

SUM(b.inadimplentes) AS inadimplentes\_outros,

SUM(b.total) AS total\_outros,

SAFE\_DIVIDE(SUM(b.inadimplentes), SUM(b.total)) AS taxa\_outros,

-- Risco relativo: taxa do quartil / taxa dos demais

SAFE\_DIVIDE(a.taxa\_inadimplencia, SAFE\_DIVIDE(SUM(b.inadimplentes), SUM(b.total))) AS risco\_relativo

FROM base a

JOIN base b

ON a.quartil\_xx != b.quartil\_xx

GROUP BY a.quartil\_xx, a.inadimplentes, a.total, a.taxa\_inadimplencia

)

SELECT

quartil\_base,

ROUND(taxa\_base, 4) AS taxa\_quartil,

ROUND(taxa\_outros, 4) AS taxa\_demais,

ROUND(risco\_relativo, 2) AS risco\_relativo

FROM combinado

ORDER BY quartil\_base;

planilha unificada riscos

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.RR\_combinado` AS

SELECT 'age' AS variavel, quartil\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_age`

UNION ALL

SELECT 'salary' AS variavel, quartil\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_salary`

UNION ALL

SELECT 'dependents' AS variavel, quartil\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_dependents`

UNION ALL

SELECT 'days' AS variavel, quartil\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_days`

UNION ALL

SELECT 'loan' AS variavel, quartil\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_loan`

UNION ALL

SELECT 'lines' AS variavel, quartil\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_lines`

UNION ALL

SELECT 'debt' AS variavel, quartil\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_debt`;

risco por cliente

-- Relacionando as informações da tabela de risco relativo por quartil das variáveis com as informações dos clientes da minha tabela geral

SELECT

u.user\_id,

u.age,

u.quartil\_age,

rr\_age.risco\_relativo AS risco\_age,

u.last\_month\_salary,

u.quartil\_salary,

rr\_salary.risco\_relativo AS risco\_salary,

u.number\_dependents,

u.quartil\_dependents,

rr\_dependents.risco\_relativo AS risco\_dependents,

u.debt\_ratio,

u.quartil\_debt,

rr\_debt.risco\_relativo AS risco\_debt,

u.using\_lines\_not\_secured\_personal\_assets,

u.quartil\_lines,

rr\_lines.risco\_relativo AS risco\_lines,

u.loans\_total,

u.quartil\_loan,

rr\_loan.risco\_relativo AS risco\_loan,

u.more\_90\_days\_overdue,

u.quartil\_days,

rr\_days.risco\_relativo AS risco\_days

FROM `projeto3-469700.dataset\_banco.tabela\_QUARTIS` u

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado` rr\_age

ON rr\_age.variavel = 'age' AND rr\_age.quartil\_base = u.quartil\_age

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado` rr\_salary

ON rr\_salary.variavel = 'salary' AND rr\_salary.quartil\_base = u.quartil\_salary

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado` rr\_dependents

ON rr\_dependents.variavel = 'dependents' AND rr\_dependents.quartil\_base = u.quartil\_dependents

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado` rr\_debt

ON rr\_debt.variavel = 'debt' AND rr\_debt.quartil\_base = u.quartil\_debt

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado` rr\_lines

ON rr\_lines.variavel = 'lines' AND rr\_lines.quartil\_base = u.quartil\_lines

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado` rr\_loan

ON rr\_loan.variavel = 'loan' AND rr\_loan.quartil\_base = u.quartil\_loan

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado` rr\_days

ON rr\_days.variavel = 'days' AND rr\_days.quartil\_base = u.quartil\_days;

-- planilha para incluir também a coluna default\_flag

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.RR\_por\_cliente\_com\_flag` AS

SELECT

rr.\*,

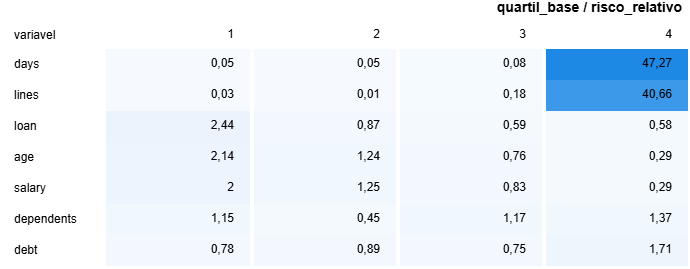
tq.default\_flag

FROM `projeto3-469700.dataset\_banco.RR\_por\_cliente` rr

LEFT JOIN `projeto3-469700.dataset\_banco.tabela\_QUARTIS` tq

ON rr.user\_id = tq.user\_id;

avaliação de risco



| variável | quartil\_base | risco\_relativo |
| --- | --- | --- |
| age | 4 | 0,29 |
| age | 3 | 0,76 |
| age | 2 | 1,24 |
| age | 1 | 2,14 |
| days | 4 | 47,27 |
| days | 3 | 0,08 |
| days | 2 | 0,05 |
| days | 1 | 0,05 |
| debt | 4 | 1,71 |
| debt | 3 | 0,75 |
| debt | 2 | 0,89 |
| debt | 1 | 0,78 |
| dependents | 4 | 1,37 |
| dependents | 3 | 1,17 |
| dependents | 2 | 0,45 |
| dependents | 1 | 1,15 |
| lines | 4 | 40,66 |
| lines | 3 | 0,18 |
| lines | 2 | 0,01 |
| lines | 1 | 0,03 |
| loan | 4 | 0,58 |
| loan | 3 | 0,59 |
| loan | 2 | 0,87 |
| loan | 1 | 2,44 |
| salary | 4 | 0,29 |
| salary | 3 | 0,83 |
| salary | 2 | 1,25 |
| salary | 1 | 2 |

| **Avaliação de taxa de risco relativo por quartil e variável** | | | |
| --- | --- | --- | --- |
| **VARIÁVEL** | **QUARTIL** | **TAXA** | **CONCLUSÕES** |
| Age | 1 (idade até 41 anos) | 2,14 | Clientes com idade de até 41 anos possuem um risco 2x maior de se tornarem inadimplentes |
| Days | 4 (possui clientes sem atraso) | 47,27 | Como a esmagadora maioria dos clientes nunca atrasou mais que 90 dias, tenho em todos os quartis dessa variável clientes que nunca atrasaram. Com isso, considero que não seja uma boa variável para usar na análise |
| Debt | 4 (taxa a partir de 0,48) | 1,71 | Clientes com taxa de endividamento a partir de 0,48 possuem um risco 1,71x maior de se tornarem inadimplentes |
| Dependents | 4 (a partir de 2 filhos) | 1,37 | Clientes que possuem a partir de 2 dependentes têm um risco 1,37 maior de se tornarem inadimplentes |
| Lines | 4 (a partir de 0,57) | 40,66 | Clientes que utilizam a partir de 0,57 do seu limite de crédito, em linhas que não são garantidas por bens pessoais possuem um risco 40x maior de se tornarem inadimplentes. Taxa muito alta em relação às demais variáveis. Vou desconsiderar no cálculo de score único por cliente. |
| Loan | 1 (até 5 empréstimos) | 2,44 | Dado incoerente, não faz sentido que os clientes com menor número de empréstimos tenham um maior risco de inadimplência. |
| Salary | 1 (salário até 3400) | 2 | Clientes cujo último salário foi até $3400,00 possuem um risco 2x maior de se tornarem inadimplentes |

Para essa planilha trouxe apenas o quartil com a maior taxa de risco relativo de cada variável avaliada.

As linhas em verde representam as variáveis consideradas para cálculo de score que será realizado na etapa seguinte enquanto as linhas em cinza representam as variáveis desconsideradas para o cálculo.

Nota: Me parece que a divisão por quartis não é a melhor forma de segmentar os dados, pois fornece um ponto de corte que reflete a quantidade de dados ordenados e não considera os valores dos dados em si. Dessa forma, clientes com a mesma característica para uma dada variável (por exemplo, clientes que nunca atrasaram o pagamento) podem ficar em quartis distintos e com isso a classificação não é feita da mesma forma para os dois.

Acho que a melhor forma de segmentar os clientes seria avaliando a faixa de variação dos dados, eliminando outliers e criando faixas estratégicas que se adequem aos dados.

score

-- Gerando o score dos clientes com base no risco das variáveis idade, último salário, número de dependentes e taxa de endividamento

SELECT

user\_id,

risco\_age,

risco\_salary,

risco\_dependents,

risco\_debt,

default\_flag,

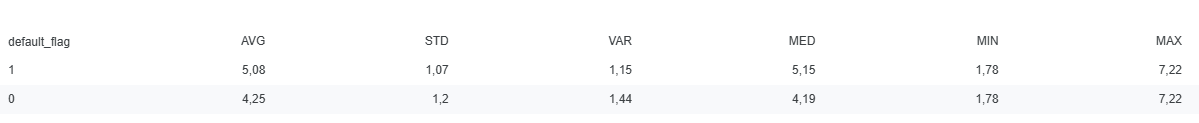
-- Score simples

(risco\_age + risco\_salary + risco\_dependents + risco\_debt) AS score\_risco\_simples,

FROM `projeto3-469700.dataset\_banco.RR\_por\_cliente\_com\_flag`;

Como eu defini o score de corte?

Olhando para a coluna default\_flag, avaliei a distribuição do meu score de risco para os inadimplentes (default \_flag =1).



Como a média de score para inadimplente é 5 e o desvio padrão é 1, considerei que 6 seria um bom número já que considera a média mais o desvio padrão dos dados para fornecer um ponto de corte.

Classifiquei meus clientes em bons ou maus pagadores com base nesse score e transformei essa classificação em uma variável binária onde 0 indica bons pagadores e 1 indica maus pagadores.

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.perfil` AS

SELECT

\*,

CASE

WHEN score\_risco\_simples > 6 THEN 'mau pagador'

ELSE 'bom pagador'

END AS classificacao\_pagador,

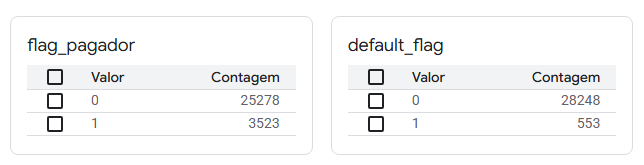
CASE

WHEN classificacao\_pagador = 'mau pagador' THEN 1

ELSE 0

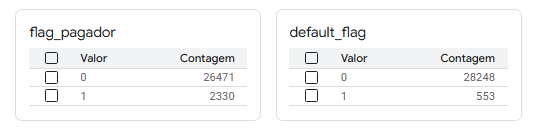
END AS flag\_pagador

FROM `projeto3-469700.dataset\_banco.score\_RR`;



Considerei os valores da média e desvio com duas casas decimais de precisão para ver se isso refletia no desempenho do meu modelo e ajustei o score de corte para 6,1.

Obtive a seguinte distribuição:



matriz de confusão

Atualizei minha tabela com uma coluna que classifica meus dados comparando os valores das colunas default\_flag e flag\_pagador. Utilizei essa coluna para identificar como os dados estão distribuídos na matriz de confusão.

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.perfil` AS

SELECT

\*,

CASE

WHEN default\_flag = 1 AND flag\_pagador = 1 THEN '+1'

WHEN default\_flag = 0 AND flag\_pagador = 0 THEN '+0'

WHEN default\_flag = 1 AND flag\_pagador = 0 THEN '-0'

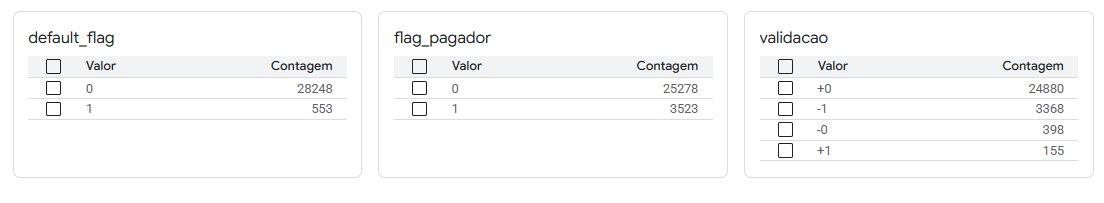
WHEN default\_flag = 0 AND flag\_pagador = 1 THEN '-1'

ELSE NULL

END AS validacao

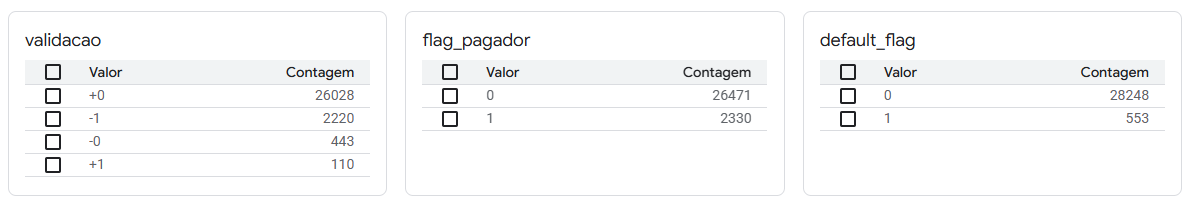
FROM `projeto3-469700.dataset\_banco.perfil`;

Score de corte 6.0:



| SCORE DE CORTE 6.0 | | MODELO | |
| --- | --- | --- | --- |
| BOM PAGADOR | MAU PAGADOR |
| DEFAULT FLAG | BOM PAGADOR | (VP) 24880 | (FN) 3368 |
| MAU PAGADOR | (FP) 398 | (VN) 155 |

Score de corte ajustado para 6,1:



| SCORE DE CORTE 6.1 | | MODELO | |
| --- | --- | --- | --- |
| BOM PAGADOR | MAU PAGADOR |
| DEFAULT FLAG | BOM PAGADOR | (VP) 26028 | (FN) 2220 |
| MAU PAGADOR | (FP) 443 | (VN) 110 |

calculo de desempenho

WITH contagem AS (

SELECT

COUNT(\*) AS total,

COUNTIF(validacao = '+1') AS vn, -- Verdadeiro Negativo

COUNTIF(validacao = '+0') AS vp, -- Verdadeiro Positivo

COUNTIF(validacao = '-1') AS fn, -- Falso Negativo

COUNTIF(validacao = '-0') AS fp -- Falso Positivo

FROM `projeto3-469700.dataset\_banco.perfil`

)

SELECT

total,

vp,

vn,

fp,

fn,

ROUND((vp + vn) / total, 4) AS acuracia,

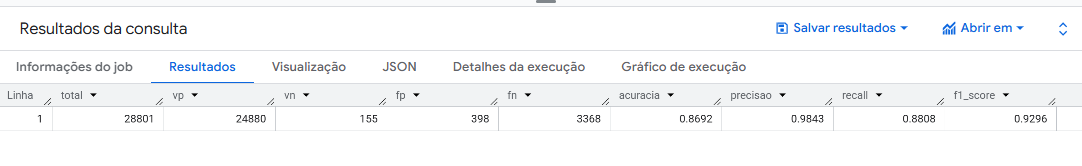
ROUND(vp / NULLIF(vp + fp, 0), 4) AS precisao,

ROUND(vp / NULLIF(vp + fn, 0), 4) AS recall,

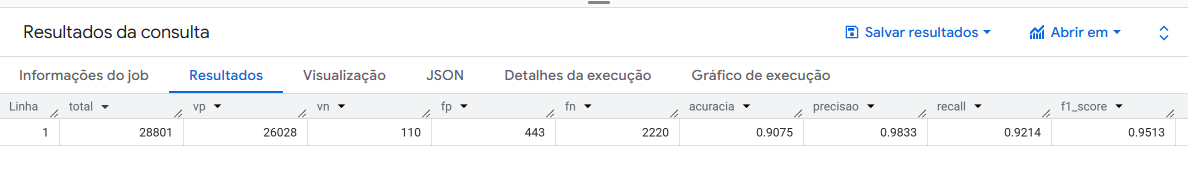
ROUND(2 \* vp / NULLIF(2 \* vp + fp + fn, 0), 4) AS f1\_score

FROM contagem;

Resultado de desempenho para modelo com score de corte 6



Resultado de desempenho para modelo com score de corte 6.1



RR por faixas

Realizei consultas para calcular, média, mediana, máximo e mínimo dos dados em cada variável para traçar uma nova classificação de faixas diferentes dos quartis e avaliar o impacto dessa abordagem no desempenho do modelo que estava baixíssimo (porque eu estava classificando errado as minhas variáveis quanto a positivo, negativo, verdadeiro e falso na hora do teste de desempenho, mas isso eu só descobri depois de tentar essa estratégia das faixas)

SELECT

MIN (using\_lines\_not\_secured\_personal\_assets) AS min\_using\_lines,

MAX (using\_lines\_not\_secured\_personal\_assets) AS max\_using\_lines,

AVG (using\_lines\_not\_secured\_personal\_assets) AS avg\_using\_lines,

FROM `projeto3-469700.dataset\_banco.tabela\_unificada`

-- recriando os quartis na forma de faixas com base no valor dos dados

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.tabela\_faixas` AS

SELECT \*,

CASE

WHEN last\_month\_salary < 2000 THEN 1

WHEN last\_month\_salary BETWEEN 2000 AND 4999 THEN 2

WHEN last\_month\_salary BETWEEN 5000 AND 10000 THEN 3

ELSE 4

END

AS faixas\_salario,

CASE

WHEN age < 40 THEN 1

WHEN age BETWEEN 40 AND 49 THEN 2

WHEN age BETWEEN 50 AND 60 THEN 3

ELSE 4

END

AS faixas\_idade,

CASE

WHEN debt\_ratio < 0.20 THEN 1

WHEN debt\_ratio BETWEEN 0.20 AND 0.35 THEN 2

WHEN debt\_ratio BETWEEN 0.36 AND 0.55 THEN 3

ELSE 4

END

AS faixas\_endividamento,

CASE

WHEN number\_dependents = 0 THEN 1

WHEN number\_dependents BETWEEN 1 AND 3 THEN 2

WHEN number\_dependents BETWEEN 4 AND 6 THEN 3

ELSE 4

END

AS faixas\_dependentes FROM `projeto3-469700.dataset\_banco.tabela\_unificada`

-- calculando taxa de ocorrência e o risco relativo para cada faixa das variáveis

-- Calculo da taxa de incidência

WITH base AS (

SELECT

faixas\_dependentes,

COUNT(\*) AS total,

COUNTIF(default\_flag = 1) AS inadimplentes,

SAFE\_DIVIDE(COUNTIF(default\_flag = 1), COUNT(\*)) AS taxa\_inadimplencia

FROM `projeto3-469700.dataset\_banco.tabela\_faixas`

GROUP BY faixas\_dependentes

),

combinado AS (

SELECT

a.faixas\_dependentes AS faixa\_base,

-- Dados do quartil avaliado

a.inadimplentes AS inadimplentes\_base,

a.total AS total\_base,

a.taxa\_inadimplencia AS taxa\_base,

-- Dados combinados dos demais quartis

SUM(b.inadimplentes) AS inadimplentes\_outros,

SUM(b.total) AS total\_outros,

SAFE\_DIVIDE(SUM(b.inadimplentes), SUM(b.total)) AS taxa\_outros,

-- Risco relativo: taxa do quartil / taxa dos demais

SAFE\_DIVIDE(a.taxa\_inadimplencia, SAFE\_DIVIDE(SUM(b.inadimplentes), SUM(b.total))) AS risco\_relativo

FROM base a

JOIN base b

ON a.faixas\_dependentes != b.faixas\_dependentes

GROUP BY a.faixas\_dependentes, a.inadimplentes, a.total, a.taxa\_inadimplencia

)

SELECT

faixa\_base,

ROUND(taxa\_base, 4) AS taxa\_quartil,

ROUND(taxa\_outros, 4) AS taxa\_demais,

ROUND(risco\_relativo, 2) AS risco\_relativo

FROM combinado

ORDER BY faixa\_base;

-- unindo todas os riscos em uma tabela

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.RR\_combinado2` AS

SELECT 'idade' AS variavel, faixa\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_idade2`

UNION ALL

SELECT 'salario' AS variavel, faixa\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_salario2`

UNION ALL

SELECT 'dependentes' AS variavel, faixa\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_dep2`

UNION ALL

SELECT 'endividamento' AS variavel, faixa\_base, taxa\_quartil, taxa\_demais, risco\_relativo

FROM `projeto3-469700.dataset\_banco.RR\_endiv2`;

-- criando uma tabela que associa o risco ao id do cliente

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.score\_RR` AS

SELECT

u.user\_id,

u.age,

u.faixas\_idade,

rr\_age.risco\_relativo AS risco\_age,

u.last\_month\_salary,

u.faixas\_salario,

rr\_salary.risco\_relativo AS risco\_salary,

u.number\_dependents,

u.faixas\_dependentes,

rr\_dependents.risco\_relativo AS risco\_dependents,

u.debt\_ratio,

u.faixas\_endividamento,

rr\_debt.risco\_relativo AS risco\_debt,

FROM `projeto3-469700.dataset\_banco.tabela\_faixas` u

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado2` rr\_age

ON rr\_age.variavel = 'idade' AND rr\_age.faixa\_base = u.faixas\_idade

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado2` rr\_salary

ON rr\_salary.variavel = 'salario' AND rr\_salary.faixa\_base = u.faixas\_salario

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado2` rr\_dependents

ON rr\_dependents.variavel = 'dependentes' AND rr\_dependents.faixa\_base = u.faixas\_dependentes

LEFT JOIN `projeto3-469700.dataset\_banco.RR\_combinado2` rr\_debt

ON rr\_debt.variavel = 'endividamento' AND rr\_debt.faixa\_base = u.faixas\_endividamento

-- gerando um score de risco único por cliente

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.JUST\_score\_RR`AS

SELECT

user\_id,

risco\_idade,

risco\_salario,

risco\_dependentes,

risco\_endividamento,

default\_flag,

-- Score simples

(risco\_idade + risco\_salario + risco\_dependentes + risco\_endividamento) AS score\_risco\_simples,

FROM `projeto3-469700.dataset\_banco.score\_RR`

-- classificando meus clientes em bom ou maus pagadores com base no score

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.perfil2` AS

SELECT

\*,

CASE

WHEN score\_risco\_simples > 5.55 THEN 'mau pagador'

ELSE 'bom pagador'

END AS class\_pagador,

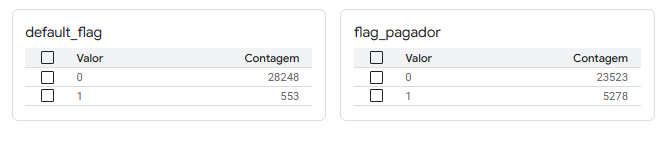
CASE

WHEN score\_risco\_simples > 5.55 THEN 1

ELSE 0

END AS flag\_pagador

FROM `projeto3-469700.dataset\_banco.JUST\_score\_RR`



-- validação do modelo

CREATE OR REPLACE TABLE `projeto3-469700.dataset\_banco.perfil2` AS

SELECT

\*,

CASE

WHEN default\_flag = 1 AND flag\_pagador = 1 THEN '+1'

WHEN default\_flag = 0 AND flag\_pagador = 0 THEN '+0'

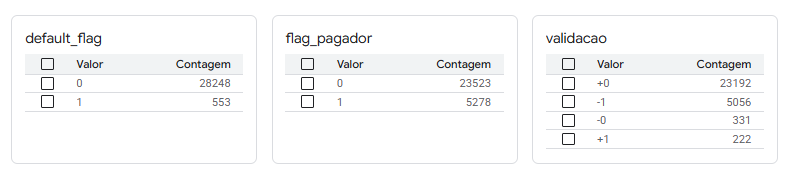
WHEN default\_flag = 1 AND flag\_pagador = 0 THEN '-0'

WHEN default\_flag = 0 AND flag\_pagador = 1 THEN '-1'

ELSE NULL

END AS validacao

FROM `projeto3-469700.dataset\_banco.perfil2`



-- cálculo de desempenho do modelo

WITH contagem AS (

SELECT

COUNT(\*) AS total,

COUNTIF(validacao = '+1') AS vn, -- Verdadeiro Negativo

COUNTIF(validacao = '+0') AS vp, -- Verdadeiro Positivo

COUNTIF(validacao = '-1') AS fn, -- Falso Negativo

COUNTIF(validacao = '-0') AS fp -- Falso Positivo

FROM `projeto3-469700.dataset\_banco.perfil2`

)

SELECT

total,

vp,

vn,

fp,

fn,

ROUND((vp + vn) / total, 4) AS acuracia,

ROUND(vp / NULLIF(vp + fp, 0), 4) AS precisao,

ROUND(vp / NULLIF(vp + fn, 0), 4) AS recall,

ROUND(2 \* vp / NULLIF(2 \* vp + fp + fn, 0), 4) AS f1\_score

FROM contagem;

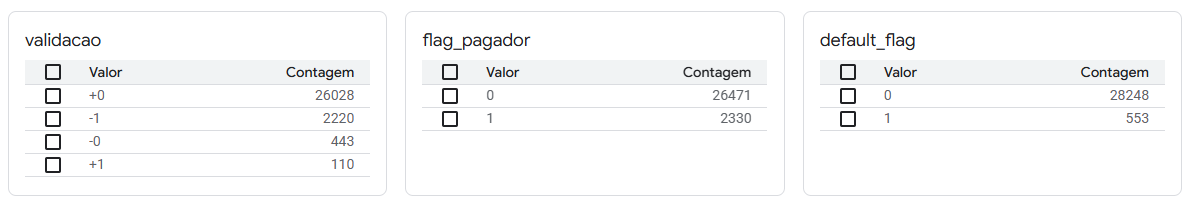


A utilização de faixas não promoveu melhora no desempenho do meu modelo. De encontro ao resultado esperado, houve redução na acurácia, recall e F1\_score. Enquanto a precisão se manteve.

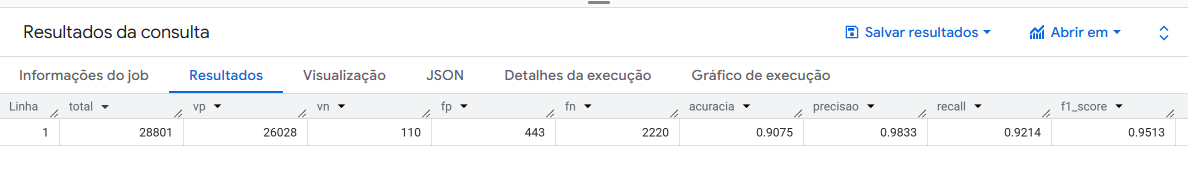
comparando os perfis

perfil

Cálculo de score de risco considerando 4 variáveis (idade, salário, dependentes e endividamento). Valor de corte do score: 6.0

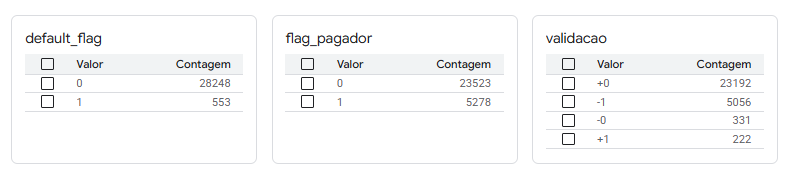


desempenho:



perfil2

Cálculo de score de risco considerando 4 variáveis (idade, salário, dependentes e endividamento) na forma de faixas definidas com base nos dados . Vide aba RR por faixas do documento

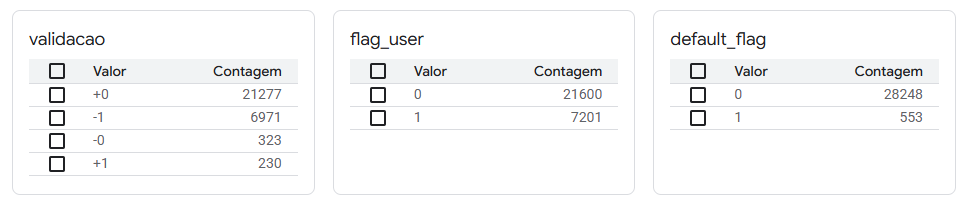


desempenho:



perfil3

Realizei também o cálculo do score de cliente utilizando apenas a variável idade. Para isso, considerei o score\_age como meu score de risco e mantive o valor de corte em 6,0. Como é possível observar, este modelo foi o que obteve o pior desempenho. Com isso, há evidências de que quanto mais variáveis consideradas para análise de risco, mais refinado e preciso é o modelo gerado.



Desempenho:

