



Projeto 2 - Risco Relativo



Ficha Técnica:

1. Objetivo da análise

No atual cenário financeiro, a diminuição das taxas de juros tem gerado um notável aumento na demanda por crédito no banco "Super Caja". No entanto, essa crescente demanda tem sobrecarregado a equipe de análise de crédito, que atualmente está imersa em um processo manual ineficiente e demorado para avaliar as inúmeras solicitações de empréstimo.

Diante desse desafio, propõe-se uma solução inovadora: a automatização do processo de análise por meio de técnicas avançadas de análise de dados. O objetivo principal é melhorar a eficiência e a precisão na avaliação do risco de crédito, permitindo ao banco tomar decisões informadas sobre a concessão de crédito e reduzir o risco de empréstimos não reembolsáveis. Esta proposta também destaca a integração de uma métrica existente de pagamentos em atraso, fortalecendo assim a capacidade do modelo

Análise Geral

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

Arquivo	Variável	Descrição
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 state = imóveis, others= outros)

Arquivo	Variável	Descrição
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)

Arquivo	Variável	Descrição
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)
	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)

Hipótese

1 - Os mais jovens correm um risco maior de não pagamento:

Conclusão:

Hipótese Confirmada! Ao lado podemos ver que pessoas com idade entre 21 e 41 e entre 41 e 52 anos possuem um perfil de mal pagador, confirmando a hipótese de que quanto mais jovem maior é o risco de inadimplência.

2 - Pessoas com mais empréstimos ativos correm maior risco de serem maus pagadores.

Conclusão:

Hipótese Refutada! Pessoas classificadas como maus pagadores são aquelas com uma quantidade de empréstimos ativos entre 1 e 8.

3 - Pessoas que atrasam seus pagamentos por mais de 90 dias correm maior risco de serem maus pagadores.

Conclusão:

Hipótese Confirmada! Segundo a análise de risco relativo, pessoas que atrasam seus pagamentos por mais de 90 dias tendem a ser maus pagadores.

Análises Estatística Exploratória

Em nossa base de dados, podemos identificar que não há discrepância salarial significativa entre os gêneros presentes.

O maior desvio padrão é observado na coluna "D-P_endividamento", com um valor de 2.013,36. Isto sugere uma grande variabilidade nos níveis de endividamento dentro do conjunto de dados analisados.

O segundo maior desvio padrão é para "D-P_idade" (14,78), indicando também uma considerável variação nas idades dos indivíduos ou itens analisados.

"D-P_empréstimos" têm um desvio padrão de 5,1, o que pode indicar uma dispersão moderada nos valores dos empréstimos.

"D-P_dependentes" apresenta um desvio padrão relativamente baixo (1,11), sugerindo menor variabilidade no número de dependentes.

O menor desvio padrão é observado em "D-P_90dias" (0,5), o que pode indicar uma distribuição mais uniforme ou concentrada para esta variável.

Baseado na análise da tabela de desvios padrão, temos uma visão geral da variabilidade de diferentes aspectos financeiros e demográficos. O endividamento mostra a maior dispersão, seguido pela idade, enquanto o número de dependentes e a variável de 90 dias apresentam menor variabilidade.

Conclusão

Esses insights destacam a complexidade das relações entre os diferentes fatores financeiros e comportamentais dos clientes, sugerindo a necessidade de estratégias personalizadas para cada segmento, visando melhor gestão e mitigação de riscos.

Estas análises fornecem insights valiosos para o Banco Super Caja, permitindo que a instituição:

Aprimore suas estratégias de gerenciamento de risco.

Desenvolvam abordagens personalizadas no relacionamento com diferentes perfis de clientes.

Implemente medidas preventivas para reduzir a inadimplência.

Otimize seus processos de concessão de crédito.

Em suma, esta análise se apresenta como uma ferramenta poderosa para a tomada de decisões baseadas em dados, possibilitando uma melhor gestão do negócio e mitigação de riscos financeiros para o Banco Super Caja.

Análises Futuras

Seria valioso incluir na base de dados informações como o estado civil, o nível de escolaridade e os salários dos clientes nos últimos três meses. Esses dados permitiriam verificar se há variações nos salários a cada mês.

Além disso, manter a base de dados atualizada continuamente é fundamental para aprofundar a análise e desenvolver modelos de Machine Learning mais eficazes.

2. Tomada de decisões

Para uma análise bancária de risco relativo para conceder empréstimos, geralmente é preferível utilizar um desvio padrão mais baixo.

Limpeza dos dados:

Tabela default, renomeei as colunas, utilizei CASE para substituir valores da coluna default_flag e renomeei para classificação de inadimplência.

```
##Limpeza das tabelas##
#Substitui os nomes das colunas da tabela DEFAULT, criar uma nova coluna com
adimplentes e inadimplentes
CREATE OR REPLACE TABLE
  `riscorelativo3.Projeto03.default` AS
WITH
  defaultt AS (
    SELECT
      user_id AS id_usuario,
      CASE
        WHEN default_flag >= 1 THEN 'INADIMPLENTE'
        ELSE 'ADIMPLENTE'
      END
    AS adimplentes_inadimplentes,
      default_flag AS classificacao_de_inadimplencia
    FROM
      `riscorelativo3.Projeto03.default` )
SELECT
  *
FROM
  defaultt;
```

Tabela loans_detail, renomeei as colunas e utilizei CAST para substituir valores de FLOAT para INT64.

```
#Substitui os nomes das colunas da tabela detail, alterar as colunas float para
INT64
CREATE OR REPLACE TABLE
  `riscorelativo3.Projeto03.loans_detail` AS
WITH
  detail AS (
    SELECT
```

```

        user_id AS id_usuario,
        more_90_days_overdue AS atraso_superior_90_dias,
        CAST(using_lines_not_secured_personal_assets AS INT64) AS
linhas_nao_protegidas_por_bens_pessoais,
        number_times_delayed_payment_loan_30_59_days AS atraso_30_59_dias,
        number_times_delayed_payment_loan_60_89_days AS atraso_60_89_dias,
        CAST(debt_ratio AS INT64) AS taxa_de_endividamento
    FROM
        `riscorelativo3.Projeto03.loans_detail` )
SELECT
    *
FROM
    detail;

```

Tabela loans_outstanding, renomeei as colunas e utilizei CASE para substituir variáveis que estavam fora do padrão: 'real estate', 'other', 'others', 'REAL ESTATE', 'Real Estate', 'Other', 'OTHER'. Substituídas por tipo de empréstimo IMÓVEL ou OUTROS.

#Substitui os nomes das colunas da tabela outstanding, alterando os dados da coluna loan_type para os tipos de empréstimos (imóvel ou outros).

```

CREATE OR REPLACE TABLE
    `riscorelativo3.Projeto03.loans_outstanding` AS
WITH
    outstanding AS (
    SELECT
        user_id AS id_usuario,
        loan_id AS id_emprestimo,
        CASE
            WHEN(loan_type) = 'real estate' THEN 'IMÓVEL'
            WHEN(loan_type) = 'other' THEN 'OUTRO'
            WHEN(loan_type) = 'others' THEN 'OUTRO'
            WHEN(loan_type) = 'REAL ESTATE' THEN 'IMÓVEL'
            WHEN(loan_type) = 'Real Estate' THEN 'IMÓVEL'
            WHEN(loan_type) = 'Other' THEN 'OUTRO'
            WHEN(loan_type) = 'OTHER' THEN 'OUTRO'
            ELSE 'SEM_EMPRESTIMO'
        END
        AS tipo_de_emprestimo
    FROM
        `riscorelativo3.Projeto03.loans_outstanding` )
SELECT
    *
FROM
    outstanding;

```

Tabela user_info, substitui os nomes das colunas da tabela user_info, alterar os dados da coluna last_month_salary usei o COALESCE para substituir pela mediana de salário e na coluna number_dependent por 0 onde for nulo. Na primeira subquery eu criei uma nova variavel chamada total_de_emprestimos_por_id com base na tabela loans_outstanding e usei o COALESCE para substituir os valores nulos pelo cálculo da mediana criando uma nova coluna chamada qtd_emprestimos_por_usuario e na segunda subquery alterei o tipo de dado da coluna ultimo_salario para INT64. e um LEFT JOIN ao final para unir as duas tabelas

```
# Substitui os nomes das colunas da tabela user_info.
CREATE OR REPLACE TABLE
  `riscorelativo3.Projeto03.user_info` AS
WITH
  user AS (
    SELECT
      user_id AS id_usuario,
      age AS idade,
      UPPER(sex) AS genero,
      COALESCE(last_month_salary, (
        SELECT
          APPROX_QUANTILES(last_month_salary, 100)[
            OFFSET
              (50)]
        FROM
          `riscorelativo3.Projeto03.user_info`
        WHERE
          last_month_salary IS NOT NULL)) AS ultimo_salario,
      COALESCE(number_dependents, 0) AS qtd_dependentes
    FROM
      `riscorelativo3.Projeto03.user_info`
  ),
  Total_emprestimos AS (
    SELECT
      id_usuario,
      COUNT(DISTINCT id_emprestimo) AS total_de_emprestimos_por_id
    FROM `riscorelativo3.Projeto03.loans_outstanding` AS outs
    GROUP BY id_usuario
  )
SELECT
  user.id_usuario,
  user.idade,
  user.genero,
  CAST(user.ultimo_salario AS INT64) AS ultimo_salario,
  user.qtd_dependentes,
  COALESCE(te.total_de_emprestimos_por_id, (
```

```

SELECT
    APPROX_QUANTILES(te.total_de_emprestimos_por_id, 100)[
        OFFSET
            (50)]
FROM
    Total_emprestimos AS te
WHERE
    te.total_de_emprestimos_por_id IS NOT NULL)) AS
qtd_emprestimos_por_usuario
FROM
    user
LEFT JOIN Total_emprestimos AS te
ON user.id_usuario = te.id_usuario;

```

```

# LEFT JOIN para unir as tabelas limpas e criar uma nova tabela (risco_relativo)
CREATE TABLE `riscorelativo3.Projeto03.risco_relativo` AS
WITH risco_relativo AS (
    SELECT
        user.id_usuario,
        user.idade,
        user.genero,
        user.ultimo_salario,
        user.qtd_emprestimos_por_usuario,
        qtd_dependentes,
        dft.adimplentes_inadimplentes,
        dft.classificacao_de_inadimplencia,
        dtl.atraso_superior_90_dias,
        dtl.linhas_nao_protegidas_por_bens_pessoais,
        dtl.atraso_30_59_dias,
        dtl.atraso_60_89_dias,
        dtl.taxa_de_endividamento,
    FROM
        `riscorelativo3.Projeto03.user_info` AS user
    LEFT JOIN `riscorelativo3.Projeto03.default` AS dft
        ON user.id_usuario = dft.id_usuario
    LEFT JOIN `riscorelativo3.Projeto03.loans_detail` AS dtl
        ON user.id_usuario = dtl.id_usuario
)
SELECT * FROM risco_relativo;

```



```

##verificar Outliers##
# Max Min, Avg, Med e Desvio Padrão.
WITH Total_emprestimos AS (
    SELECT
        id_usuario,
        COUNT(DISTINCT id_emprestimo) AS total_de_emprestimos_por_id
    FROM `riscorelativo3.Projeto03.loans_outstanding`
    GROUP BY id_usuario
)
SELECT
    MIN(total_de_emprestimos_por_id) AS min_total_por_id,
    MAX(total_de_emprestimos_por_id) AS max_total_por_id,
    AVG(total_de_emprestimos_por_id) AS media_total_por_id,
    APPROX_QUANTILES(total_de_emprestimos_por_id, 100)[SAFE_ORDINAL(50)] AS
    mediana_total_por_id,
    STDDEV(total_de_emprestimos_por_id) AS desvio_total_por_id,
    FROM Total_emprestimos;

```

```

# Max Min, Avg, Med e Desvio Padrão.
SELECT
    MIN(idade) AS min_idade,
    MAX(idade) AS max_idade,
    AVG(idade) AS media_idade,
    APPROX_QUANTILES(idade, 100)[SAFE_ORDINAL(50)] AS mediana_idade,
    STDDEV(idade) AS desvio_idade,

    MIN(ultimo_salario) AS min_salario,
    MAX(ultimo_salario) AS max_salario,
    AVG(ultimo_salario) AS media_salario,
    APPROX_QUANTILES(ultimo_salario, 100)[SAFE_ORDINAL(50)] AS mediana_salario,
    STDDEV(ultimo_salario) AS desvio_salario,

    MIN(qtd_dependentes) AS min_dependentes,
    MAX(qtd_dependentes) AS max_dependentes,
    AVG(qtd_dependentes) AS media_dependentes,
    APPROX_QUANTILES(qtd_dependentes, 100)[SAFE_ORDINAL(50)] AS mediana_dependentes,
    STDDEV(qtd_dependentes) AS desvio_dependentes,

    MIN(qtd_emprestimos_por_usuario) AS min_emprestimo,
    MAX(qtd_emprestimos_por_usuario) AS max_emprestimo,
    AVG(qtd_emprestimos_por_usuario) AS media_emprestimo,

```

```

APPROX_QUANTILES(qtd_emprestimos_por_usuario, 100)[SAFE_ORDINAL(50)] AS
mediana_emprestimo,
STDDEV(qtd_emprestimos_por_usuario) AS desvio_emprestimo,

MIN(atraso_superior_90_dias) AS min_atraso_90,
MAX(atraso_superior_90_dias) AS max_atraso_90,
AVG(atraso_superior_90_dias) AS media_atraso_90,
APPROX_QUANTILES(atraso_superior_90_dias, 100)[SAFE_ORDINAL(50)] AS
mediana_atraso_90,
STDDEV(atraso_superior_90_dias) AS desvio_atraso_90,

MIN(atraso_30_59_dias) AS min_atraso_30_59,
MAX(atraso_30_59_dias) AS max_atraso_30_59,
AVG(atraso_30_59_dias) AS media_atraso_30_59,
APPROX_QUANTILES(atraso_30_59_dias, 100)[SAFE_ORDINAL(50)] AS
mediana_atraso_30_59,
STDDEV(atraso_30_59_dias) AS desvio_atraso_30_59,

MIN(atraso_60_89_dias) AS min_atraso_60_89,
MAX(atraso_60_89_dias) AS max_atraso_60_89,
AVG(atraso_60_89_dias) AS media_atraso_60_89,
APPROX_QUANTILES(atraso_60_89_dias, 100)[SAFE_ORDINAL(50)] AS
mediana_atraso_60_89,
STDDEV(atraso_60_89_dias) AS desvio_atraso_60_89,

MIN(taxa_de_endividamento) AS min_tx_endividamento,
MAX(taxa_de_endividamento) AS max_tx_endividamento,
AVG(taxa_de_endividamento) AS media_tx_endividamento,
APPROX_QUANTILES(taxa_de_endividamento, 100)[SAFE_ORDINAL(50)] AS
mediana_tx_endividamento,
STDDEV(taxa_de_endividamento) AS desvio_tx_endividamento

FROM `riscorelativo3.Projeto03.risco_relativo`

```

Tabela risco_relativo_ optei por excluir valores acima de 96 na tabela.

```

# Identifiquei o número 98 nas três colunas de atraso de pagamento em um total
de 63 usuários, sendo aproximadamente 0,175% do total da nossa base de 36.000
usuários.

# atraso superior a 90 = valor máximo é de 15 vezes
# atraso entre 30 e 59 dias = valor máximo é de 11 vezes
# atraso entre 60 e 89 dias = valor máximo é de 7 vezes
CREATE TABLE `riscorelativo3.Projeto03.risco_relativo_` AS (

```

```

SELECT
id_usuario,
idade,
genero,
ultimo_salario,
qtd_dependentes,
qtd_emprestimos_por_usuario,
adimplentes_inadimplentes,
classificacao_de_inadimplencia,
linhas_nao_protegidas_por_bens_pessoais,
taxa_de_endividamento,
atraso_30_59_dias,
atraso_60_89_dias,
atraso_superior_90_dias
FROM `riscorelativo3.Projeto03.risco_relativo`
WHERE atraso_superior_90_dias <= 70
      AND atraso_30_59_dias <= 70
      AND atraso_60_89_dias <= 70
)

```

Desvio padrão da tabela risco_relativo_ e o mais recomendado para uso em caso do banco, ou seja desconsidere os valores 98 contidos nas colunas de atrasos.

```

SELECT
MIN(atraso_superior_90_dias) AS min_atraso_90,
MAX(atraso_superior_90_dias) AS max_atraso_90,
AVG(atraso_superior_90_dias) AS media_atraso_90,
APPROX_QUANTILES(atraso_superior_90_dias, 100)[SAFE_ORDINAL(50)] AS
mediana_atraso_90,
STDDEV(atraso_superior_90_dias) AS desvio_atraso_90,

MIN(atraso_30_59_dias) AS min_atraso_30_59,
MAX(atraso_30_59_dias) AS max_atraso_30_59,
AVG(atraso_30_59_dias) AS media_atraso_30_59,
APPROX_QUANTILES(atraso_30_59_dias, 100)[SAFE_ORDINAL(50)] AS
mediana_atraso_30_59,
STDDEV(atraso_30_59_dias) AS desvio_atraso_30_59,

MIN(atraso_60_89_dias) AS min_atraso_60_89,
MAX(atraso_60_89_dias) AS max_atraso_60_89,
AVG(atraso_60_89_dias) AS media_atraso_60_89,

```

```

APPROX_QUANTILES(atraso_60_89_dias, 100)[SAFE_ORDINAL(50)] AS
mediana_atraso_60_89,
STDDEV(atraso_60_89_dias) AS desvio_atraso_60_89,
FROM `riscorelativo3.Projeto03.risco_relativo_`

```

```

# CORRE para verificar a correlação entre as variáveis.
SELECT
CORR(idade, classificacao_de_inadimplencia) AS hipotese1,
CORR(qtd_emprestimos_por_usuario, classificacao_de_inadimplencia) AS hipotese2,
CORR(atraso_superior_90_dias, classificacao_de_inadimplencia) AS hipotese3,
CORR(atraso_superior_90_dias, atraso_30_59_dias) AS exemplo,
CORR(qtd_emprestimos_por_usuario, idade) AS emprestimo_idade,
CORR(qtd_emprestimos_por_usuario, ultimo_salario) AS emprestimo_salario,
CORR(qtd_emprestimos_por_usuario, qtd_dependentes) AS emprestimo_dependentes
FROM `riscorelativo3.Projeto03.risco_relativo_`

```

```

# Quartil e Segmentação (Bom e Mal pagador)
CREATE OR REPLACE TABLE `riscorelativo3.Projeto03.risco_relativo_quartil` AS

WITH quartil AS (
    SELECT
        id_usuario, idade, ultimo_salario, qtd_emprestimos_por_usuario,
        qtd_dependentes, classificacao_de_inadimplencia,
        linhas_nao_protegidas_por_bens_pessoais, taxa_de_endividamento,
        atraso_30_59_dias, atraso_60_89_dias, atraso_superior_90_dias,
        NTILE(4) OVER (ORDER BY idade) AS quartil_idade,
        NTILE(4) OVER (ORDER BY ultimo_salario) AS quartil_salario,
        NTILE(4) OVER (ORDER BY qtd_emprestimos_por_usuario) AS quartil_emprestimos,
        NTILE(4) OVER (ORDER BY qtd_dependentes) AS quartil_dependentes,
        NTILE(4) OVER (ORDER BY classificacao_de_inadimplencia) AS
quartil_inadimplencia,
        NTILE(4) OVER (ORDER BY linhas_nao_protegidas_por_bens_pessoais) AS
quartil_bens_pessoais,
        NTILE(4) OVER (ORDER BY taxa_de_endividamento) AS quartil_endividamento,
        NTILE(4) OVER (ORDER BY atraso_30_59_dias) AS quartil_atraso_30_59,
        NTILE(4) OVER (ORDER BY atraso_60_89_dias) AS quartil_atraso_60_89,
        NTILE(4) OVER (ORDER BY atraso_superior_90_dias) AS
quartil_atraso_superior_90
    FROM `riscorelativo3.Projeto03.risco_relativo_`
),

```

```

total_classificacao AS (
    SELECT
        SUM(classificacao_de_inadimplencia) AS sum_class,
        COUNT(*) AS total
    FROM quartil
),

idade_risco_relativo AS (
    SELECT
        quartil_idade AS quartil,
        MIN(idade) AS min_idade,
        MAX(idade) AS max_idade,
        AVG(classificacao_de_inadimplencia) / ((total_classificacao.sum_class -
SUM(quartil.classificacao_de_inadimplencia)) / (total_classificacao.total -
COUNT(*))) AS riscorelativo_idade
    FROM quartil, total_classificacao
    GROUP BY quartil_idade, total_classificacao.sum_class,
total_classificacao.total
),

ultimo_salario_risco_relativo AS (
    SELECT
        quartil_salario AS quartil,
        MIN(ultimo_salario) AS min_salario,
        MAX(ultimo_salario) AS max_salario,
        AVG(classificacao_de_inadimplencia) / ((total_classificacao.sum_class -
SUM(quartil.classificacao_de_inadimplencia)) / (total_classificacao.total -
COUNT(*))) AS riscorelativo_salario
    FROM quartil, total_classificacao
    GROUP BY quartil_salario, total_classificacao.sum_class,
total_classificacao.total
),

qtd_dependentes_risco_relativo AS (
    SELECT
        quartil_dependentes AS quartil,
        MIN(qtd_dependentes) AS min_dependentes,
        MAX(qtd_dependentes) AS max_dependentes,
        AVG(classificacao_de_inadimplencia) / ((total_classificacao.sum_class -
SUM(quartil.classificacao_de_inadimplencia)) / (total_classificacao.total -
COUNT(*))) AS riscorelativo_dependentes
    FROM quartil, total_classificacao

```

```

    GROUP BY quartil_dependentes, total_classificacao.sum_class,
total_classificacao.total
),

qtd_emprestimos_risco_relativo AS (
    SELECT
        quartil_emprestimos AS quartil,
        MIN(qtd_emprestimos_por_usuario) AS min_emprestimos,
        MAX(qtd_emprestimos_por_usuario) AS max_emprestimos,
        AVG(classificacao_de_inadimplencia) / ((total_classificacao.sum_class -
SUM(quartil.classificacao_de_inadimplencia)) / (total_classificacao.total -
COUNT(*))) AS risco_relativo_emprestimo
    FROM quartil, total_classificacao
    GROUP BY quartil_emprestimos, total_classificacao.sum_class,
total_classificacao.total
),

atraso_superior_90_risco_relativo AS (
    SELECT
        quartil_atraso_superior_90 AS quartil,
        MIN(atraso_superior_90_dias) AS min_superior_90,
        MAX(atraso_superior_90_dias) AS max_superior_90,
        AVG(classificacao_de_inadimplencia) / ((total_classificacao.sum_class -
SUM(quartil.classificacao_de_inadimplencia)) / (total_classificacao.total -
COUNT(*))) AS risco_relativo_90
    FROM quartil, total_classificacao
    GROUP BY quartil_atraso_superior_90, total_classificacao.sum_class,
total_classificacao.total
),

total_linhas_risco_relativo AS (
    SELECT
        quartil_bens_pessoais AS quartil,
        MIN(linhas_ao_protegidas_por_bens_pessoais) AS min_total_linhas,
        MAX(linhas_ao_protegidas_por_bens_pessoais) AS max_total_linhas,
        AVG(classificacao_de_inadimplencia) / ((total_classificacao.sum_class -
SUM(quartil.classificacao_de_inadimplencia)) / (total_classificacao.total -
COUNT(*))) AS risco_relativo_total_linhas
    FROM quartil, total_classificacao
    GROUP BY quartil_bens_pessoais, total_classificacao.sum_class,
total_classificacao.total
),

```

```

taxa_de_endividamento_risco_relativo AS (
    SELECT
        quartil_endividamento AS quartil,
        MIN(taxa_de_endividamento) AS min_endividamento,
        MAX(taxa_de_endividamento) AS max_endividamento,
        AVG(classificacao_de_inadimplencia) / ((total_classificacao.sum_class -
SUM(quartil.classificacao_de_inadimplencia)) / (total_classificacao.total -
COUNT(*))) AS risco_relativo_taxa_endividamento
    FROM quartil, total_classificacao
    GROUP BY quartil_endividamento, total_classificacao.sum_class,
total_classificacao.total
),

atraso_30_59_dias_risco_relativo AS (
    SELECT
        quartil_atraso_30_59 AS quartil,
        MIN(atraso_30_59_dias) AS min_atraso_30_59,
        MAX(atraso_30_59_dias) AS max_atraso_30_59,
        AVG(classificacao_de_inadimplencia) / ((total_classificacao.sum_class -
SUM(quartil.classificacao_de_inadimplencia)) / (total_classificacao.total -
COUNT(*))) AS risco_relativo_30_59
    FROM quartil, total_classificacao
    GROUP BY quartil_atraso_30_59, total_classificacao.sum_class,
total_classificacao.total
),

atraso_60_89_dias_risco_relativo AS (
    SELECT
        quartil_atraso_60_89 AS quartil,
        MIN(atraso_60_89_dias) AS min_atraso_60_89,
        MAX(atraso_60_89_dias) AS max_atraso_60_89,
        AVG(classificacao_de_inadimplencia) / ((total_classificacao.sum_class -
SUM(quartil.classificacao_de_inadimplencia)) / (total_classificacao.total -
COUNT(*))) AS risco_relativo_60_89
    FROM quartil, total_classificacao
    GROUP BY quartil_atraso_60_89, total_classificacao.sum_class,
total_classificacao.total
)

SELECT
    a.quartil,
    a.riscorelativo_idade AS idade_risco_relativo,
    a.min_idade,

```

```

a.max_idade,
CASE
    WHEN a.riscorelativo_idade < 1 THEN 'Bom pagador'
    WHEN a.riscorelativo_idade > 1 THEN 'Mau pagador'
END AS idade_segmentacao,

b.riscorelativo_salario AS ultimo_salario_risco_relativo,
b.min_salario,
b.max_salario,
CASE
    WHEN b.riscorelativo_salario < 1 THEN 'Bom pagador'
    WHEN b.riscorelativo_salario > 1 THEN 'Mau pagador'
END AS ultimo_salario_segmentacao,

c.riscorelativo_dependentes AS dependentes_risco_relativo,
c.min_dependentes,
c.max_dependentes,
CASE
    WHEN c.riscorelativo_dependentes < 1 THEN 'Bom pagador'
    WHEN c.riscorelativo_dependentes > 1 THEN 'Mau pagador'
END AS num_dependentes_segmentacao,

d.risco_relativo_emprestimo AS total_emprestimos_risco_relativo,
d.min_emprestimos,
d.max_emprestimos,
CASE
    WHEN d.risco_relativo_emprestimo < 1 THEN 'Bom pagador'
    WHEN d.risco_relativo_emprestimo > 1 THEN 'Mau pagador'
END AS total_emprestimos_segmentacao,

e.risco_relativo_total_linhas AS
linhas_nao_protegidas_por_bens_pessoais_risco_relativo,
e.min_total_linhas,
e.max_total_linhas,
CASE
    WHEN e.risco_relativo_total_linhas < 1 THEN 'Bom pagador'
    WHEN e.risco_relativo_total_linhas > 1 THEN 'Mau pagador'
END AS total_linhas_segmentacao,

f.risco_relativo_taxa_endividamento AS taxa_de_endividamento_risco_relativo,
f.min_endividamento,
f.max_endividamento,
CASE

```



```

        WHEN f.risco_relativo_taxa_endividamento < 1 THEN 'Bom pagador'
        WHEN f.risco_relativo_taxa_endividamento > 1 THEN 'Mau pagador'
    END AS taxa_de_endividamento_segmentacao,

    g.risco_relativo_30_59 AS atraso_30_59_dias_risco_relativo,
    g.min_atraso_30_59,
    g.max_atraso_30_59,
    CASE
        WHEN g.risco_relativo_30_59 < 1 THEN 'Bom pagador'
        WHEN g.risco_relativo_30_59 > 1 THEN 'Mau pagador'
    END AS atraso_30_59_dias_segmentacao,

    h.risco_relativo_60_89 AS atraso_60_89_dias_risco_relativo,
    h.min_atraso_60_89,
    h.max_atraso_60_89,
    CASE
        WHEN h.risco_relativo_60_89 < 1 THEN 'Bom pagador'
        WHEN h.risco_relativo_60_89 > 1 THEN 'Mau pagador'
    END AS atraso_60_89_dias_segmentacao,

    i.risco_relativo_90 AS atraso_superior_90_dias_risco_relativo,
    i.min_superior_90,
    i.max_superior_90,
    CASE
        WHEN i.risco_relativo_90 < 1 THEN 'Bom pagador'
        WHEN i.risco_relativo_90 > 1 THEN 'Mau pagador'
    END AS atraso_superior_90_dias_segmentacao
FROM
    idade_risco_relativo a
    JOIN ultimo_salario_risco_relativo b ON a.quartil = b.quartil
    JOIN qtd_dependentes_risco_relativo c ON a.quartil = c.quartil
    JOIN qtd_emprestimos_risco_relativo d ON a.quartil = d.quartil
    JOIN total_linhas_risco_relativo e ON a.quartil = e.quartil
    JOIN taxa_de_endividamento_risco_relativo f ON a.quartil = f.quartil
    JOIN atraso_30_59_dias_risco_relativo g ON a.quartil = g.quartil
    JOIN atraso_60_89_dias_risco_relativo h ON a.quartil = h.quartil
    JOIN atraso_superior_90_risco_relativo i ON a.quartil = i.quartil
ORDER BY a.quartil;

```

3. Informações úteis

Ferramentas utilizadas

- BigQuery
- Looker Studio

Linguagens utilizadas

- SQL

Links úteis

- [BigQuery](#)
- [Looker Studio](#)
- [Base de Dados](#)
- [Vídeo de Apresentação](#)