

Análise de Risco de Crédito - NuBank

Risco de crédito está associado à possibilidade de um cliente não cumprir com as obrigações contratuais, como hipotecas, dívidas de cartão de crédito e outros tipos de empréstimos.

Minimizar o risco de inadimplência é uma grande preocupação para instituições financeiras. Por esse motivo, bancos comerciais e de investimento, fundos de capital de risco, empresas de gestão de ativos e seguradoras, para citar alguns, estão cada vez mais contando com a tecnologia para prever quais clientes são mais propensos a não honrar com suas dívidas.

Modelos de Machine Learning têm ajudado essas empresas a melhorar a precisão de suas análises de risco de crédito, fornecendo um método científico para identificar devedores em potencial com antecedência.

Neste projeto, construiremos um modelo para prever o risco de inadimplência do cliente para o Nubank, uma das maiores e importantes Fintechs brasileira.

Importação das Bibliotecas

```
In [1]: # import dos pacotes
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
# definição do estilo estético das plotagens
sns.set_style()

# filtragem de warnings
import warnings
warnings.filterwarnings('ignore')

pd.options.display.float_format = '{:20,.2f}'.format

# Display options.
pd.set_option('display.max_rows', None)
pd.set_option('display.max_info_rows', 100)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)

# Set the option to display all columns
pd.set_option('display.max_columns', None)

# VIF
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Visualization Settings
```

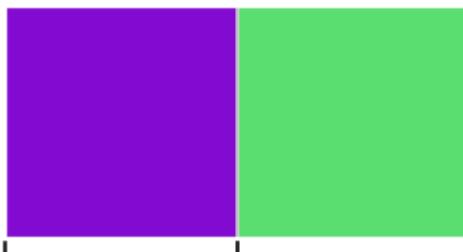
```

mpl.style.use('ggplot')
mpl.rcParams['axes.facecolor']      = 'white'
mpl.rcParams['grid.color']          = 'lightgray'
mpl.rcParams['xtick.color']         = 'black'
mpl.rcParams['ytick.color']         = 'black'
mpl.rcParams['axes.grid']           = True
mpl.rcParams['figure.dpi']          = 150

# Palette Setting
# instyle_palette = ['#830BD1', '#5ADE6F', '#57B6EB', '#FC951D', '#969393']
instyle_palette = ['#830BD1', '#5ADE6F']
sns.set_palette(sns.color_palette(instyle_palette))
sns.palplot(sns.color_palette(instyle_palette))

# sklearn
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split

```



Funções

```

In [2]: def detectar_e_substituir_outliers(df):
    """
        Função que detecta e substitui outliers por NaN com base no método IQR.
    """

    for column in df.select_dtypes(include=[np.number]).columns: # Process only
        Q1 = df[column].quantile(0.25)
        Q3 = df[column].quantile(0.75)
        IQR = Q3 - Q1

        # Define bounds for outliers
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR

        # Replace outliers with NaN
        df[column] = df[column].apply(lambda x: x if lower_bound <= x <= upper_bound else np.nan)

    print("Os valores discrepantes foram substituídos por NaN.")
    return df

# FUNÇÃO TOP 10 SCORE 3 MAIS FREQUENTES
def top_10_score_plot(data, col, title):
    top10 = data[col].value_counts().head(10)
    plt.figure(figsize=(20,5))
    plt.title(title, size=18)
    ax = sns.barplot(y=top10.index, x=top10, orient='h', color="#830BD1")
    ax.set(xlabel=None, ylabel=None)

```

```

for i in ax.containers:
    ax.bar_label(i)
plt.show()

def score_3_plot(data, col1, col2, y_label, title):
    plt.figure(figsize=(5, 3.5))
    sns.lineplot( x=col1, y=col2, data=aux1, color="purple" )

    plt.xlabel("Score 3")
    plt.ylabel(y_label)
    plt.axvline(x=data[col1].mean(), color='red', ls='--',label='Média de Score')

    plt.title(title)
    plt.legend(bbox_to_anchor=(1.0, 1), loc='upper left')
    plt.show()

```

Importação do Dataset

In [3]: `df = pd.read_csv('data/acquisition_train.csv')`

Visualização e Entendimento dos Dados

In [4]: `df.head(2)`

Out[4]:

	ids	target_default	score_1	score_2
0	343b7e7b-2cf8-e508-b8fd-0a0285af30aa	False	1Rk8w4Ucd5yR3KcqZzLdow==	IOVu8au3ISbo6+zmfnYwN

1	bc2c7502-bbad-0f8c-39c3-94e881967124	False	DGCQep2AE5QRkNCshlAlFQ==	SaamrHMo23I/3TwXOWgVz
---	--------------------------------------	-------	--------------------------	-----------------------

In [5]: `# Dataframe shape`
`print('Número de linhas do dataset: ', df.shape[0])`
`print('Número de colunas do dataset: ', df.shape[1])`

Número de linhas do dataset: 45000
Número de colunas do dataset: 43

In [6]: `# Informações sobre o dataset`
`df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45000 entries, 0 to 44999
Data columns (total 43 columns):
 #   Column                               Dtype  
 --- 
 0   ids                                  object  
 1   target_default                       object  
 2   score_1                             object  
 3   score_2                             object  
 4   score_3                             float64 
 5   score_4                             float64 
 6   score_5                             float64 
 7   score_6                             float64 
 8   risk_rate                           float64 
 9   last_amount_borrowed                float64 
 10  last_borrowed_in_months             float64 
 11  credit_limit                        float64 
 12  reason                             object  
 13  income                            float64 
 14  facebook_profile                  object  
 15  state                             object  
 16  zip                               object  
 17  channel                           object  
 18  job_name                          object  
 19  real_state                        object  
 20  ok_since                          float64 
 21  n_bankruptcies                   float64 
 22  n_defaulted_loans                 float64 
 23  n_accounts                        float64 
 24  n_issues                           float64 
 25  application_time_applied          object  
 26  application_time_in_funnel        int64   
 27  email                             object  
 28  external_data_provider_credit_checks_last_2_year float64 
 29  external_data_provider_credit_checks_last_month    int64   
 30  external_data_provider_credit_checks_last_year     float64 
 31  external_data_provider_email_seen_before          float64 
 32  external_data_provider_first_name                object  
 33  external_data_provider_fraud_score              int64   
 34  lat_lon                            object  
 35  marketing_channel                  object  
 36  profile_phone_number               object  
 37  reported_income                   float64 
 38  shipping_state                    object  
 39  shipping_zip_code                 int64   
 40  profile_tags                      object  
 41  user_agent                        object  
 42  target_fraud                     object  
dtypes: float64(18), int64(4), object(21)
memory usage: 14.8+ MB
```

- Podemos ver que algumas variáveis possuem valores ausentes. Vamos dar uma olhada mais de perto nelas.

```
In [7]: # Porcentagem de valores faltantes
print(((df.isnull().sum() / df.shape[0]) * 100).sort_values(ascending=False))
```

target_fraud	96.62
last_amount_borrowed	66.57
last_borrowed_in_months	66.57
ok_since	58.99
external_data_provider_credit_checks_last_2_year	50.28
external_data_provider_credit_checks_last_year	33.61
credit_limit	30.67
n_issues	25.65
facebook_profile	9.91
marketing_channel	7.95
job_name	7.41
target_default	7.24
external_data_provider_email_seen_before	4.96
lat_lon	3.03
user_agent	1.60
n_bankruptcies	1.55
n_defaulted_loans	1.28
reason	1.26
zip	1.25
n_accounts	1.25
channel	1.25
score_1	1.25
score_3	1.25
risk_rate	1.25
income	1.25
real_state	1.25
state	1.25
score_2	1.25
profile_tags	0.00
shipping_zip_code	0.00
shipping_state	0.00
reported_income	0.00
profile_phone_number	0.00
external_data_provider_credit_checks_last_month	0.00
external_data_provider_fraud_score	0.00
external_data_provider_first_name	0.00
score_4	0.00
score_5	0.00
score_6	0.00
email	0.00
application_time_in_funnel	0.00
application_time_applied	0.00
ids	0.00
dtype: float64	

Relatório Análise Inicial dos Dados

- O dataset possui **43** variáveis e **45.000** registros.
- Dessas **43** variáveis, **8** possuem mais de **10%** de valores missing, **target_fraud** é a variável com a maior porcentagem de valores ausentes (**96.6%**), em segundo e terceiro lugar temos as variáveis **last_amount_borrowed** e **last_borrowed_in_months** com (**66.5%**) de valores ausentes.
- A variável **target_fraud** mostra se existe ou não fraude de cartão de crédito. Nesse caso essa variável não é importante para o projeto. Portanto, esta coluna será excluída.

- As variáveis **last_amount_borrowed**, **last_borrowed_in_months**, **ok_since** e **external_data_provider_credit_checks_last_2_year** possuem mais de 50% de valores ausentes, substituir esses valores é algo difícil, manter elas pode gerar erros durante a predição. Portanto, estas colunas serão excluídas.
- As variáveis **job_name**, **external_data_provider_first_name**, **profile_phone_number** e **zip** são variáveis insignificantes para o projeto, portanto serão excluídas.
- A variável **external_data_provider_email_seen_before** possui um valor mínimo de -999.000000, certamente foi um erro, nesse caso, vamos alterar para o tipo NaN.
- A variável **reported_income** possui um valor máximo descrito como inf(infinito), nesse caso, vamos alterar para o tipo NaN.
- A variável **credit_limit** possui um valor mínimo de 0.000000, isso não existe em instituições financeiras, é obrigatório liberar um valor X de crédito para o cliente. Portanto, este valor será substituído por NaN.
- As variáveis:
 - ids**, **score_1**, **score_2**, **reason**, **facebook_profile**, **state**, **zip**, **channel**, **job_name**, **real_state**, **email**, **external_data_provider_first_name**, **lat_lon**, **marketing_channel**, **external_data_provider_email_seen_before**, **application_time_in_funnel**, **application_time_applied**, **external_data_provider_fraud_score**, **profile_phone_number**, **shipping_zip_code**, **profile_tags**, **user_agent**, e **target_fraud** são variáveis com valores não significantes para o projeto, portanto serão excluídas.

Análise Exploratória de Dados

Exclusão das Variáveis não significativas

```
In [8]: # cópia do dataframe
df2 = df.copy()
```

```
In [9]: exclude_columns = ["ids", "score_1", "score_2", "reason", "facebook_profile", "s
          "email", "external_data_provider_first_name", "external_data
          "application_time_applied", "profile_phone_number", "applica
          "profile_tags", "user_agent", "target_fraud"]

df2.drop(labels = exclude_columns, axis=1, inplace=True)
```

Alteração da Variável "credit_limit" com valor 0

- A variável **credit_limit** possui um valor mínimo de 0.000000, isso não existe em instituições financeiras, é obrigatório liberar um valor X de crédito para o cliente. Portanto, este valor será substituído por NaN.

```
In [10]: df2['credit_limit'] = df2['credit_limit'].apply(lambda x: np.nan if x == 0 else
```

Alteração da Variável "reported_income" com Valor Inf

- A variável **reported_income** possui um valor máximo descrito como inf(infinito), nesse caso, vamos alterar para o tipo NaN.

```
In [11]: df2.replace([np.inf, -np.inf], np.nan, inplace=True)
```

Divisão dos dados em treino e teste

- Antes de iniciar as transformações e pré-processamentos dos dados, é necessário dividir o dataset em treino e teste, mas porque?
 - Porque precisamos evitar o "Data Leakage". Data Leakage ocorre quando informações do conjunto de dados de teste ou validação vazam para o conjunto de treinamento durante o pré-processamento ou modelagem, ou quando informações do target vazam para as features. Nessas situações, o que vai acontecer é que você vai ver um modelo muito bom, mas isso será ilusório, pois o seu modelo "roubou" para ter o resultado bom.
- Referências sobre Data Leakage:
 - <https://www.linkedin.com/company/universidade-dos-dados/posts/?feedView=all>
 - <https://www.casadocodigo.com.br/products/livro-escd>
 - <https://estatsite.com.br/2020/12/12/data-leakage-o-erro-que-ate-os-grandes-cometem/>

```
In [12]: # Dividindo os dados primeiro
```

```
X = df2.drop(columns=['target_default'])
y = df2['target_default']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random
```

```
In [13]: train = pd.concat([X_train, y_train], axis=1)
train.reset_index(drop=True, inplace=True)
train.head()
```

	score_3	score_4	score_5	score_6	risk_rate	last_amount_borrowed	last_borrowed_ir
0	240.00	105.33	0.89	94.50	0.26		NaN
1	270.00	103.21	0.61	101.57	0.33		19,237.32
2	260.00	102.56	0.92	117.06	0.28		NaN
3	300.00	101.66	0.63	89.14	0.28		NaN
4	350.00	109.60	0.66	98.57	0.27		NaN

Describe das Variáveis

$$\text{média} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\text{desvio_padrão} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{média})^2}{n-1}}$$

\$\$\text{Mediana} =

$$\begin{cases} x_{(n+1)/2} & \text{se } n \text{ é ímpar} \\ \frac{x_{n/2} + x_{(n/2)+1}}{2} & \text{se } n \text{ é par} \end{cases}$$

In [14]: `train.describe().T`

Out[14]:

		count	mean
	score_3	33,325.00	347.83
	score_4	33,750.00	100.00
	score_5	33,750.00	0.50
	score_6	33,750.00	99.89
	risk_rate	33,325.00	0.30
	last_amount_borrowed	11,239.00	13,795.28
	last_borrowed_in_months	11,239.00	41.84
	credit_limit	19,505.00	41,384.85
	income	33,325.00	71,447.46
	ok_since	13,877.00	35.00
	n_bankruptcies	33,232.00	0.08
	n_defaulted_loans	33,319.00	0.00
	n_accounts	33,325.00	10.71
	n_issues	25,107.00	11.09
	external_data_provider_credit_checks_last_2_year	16,795.00	0.00
	external_data_provider_credit_checks_last_month	33,750.00	1.50
	external_data_provider_credit_checks_last_year	22,422.00	0.50
	reported_income	33,692.00	9,997,678,580,509.43
			251,877,19

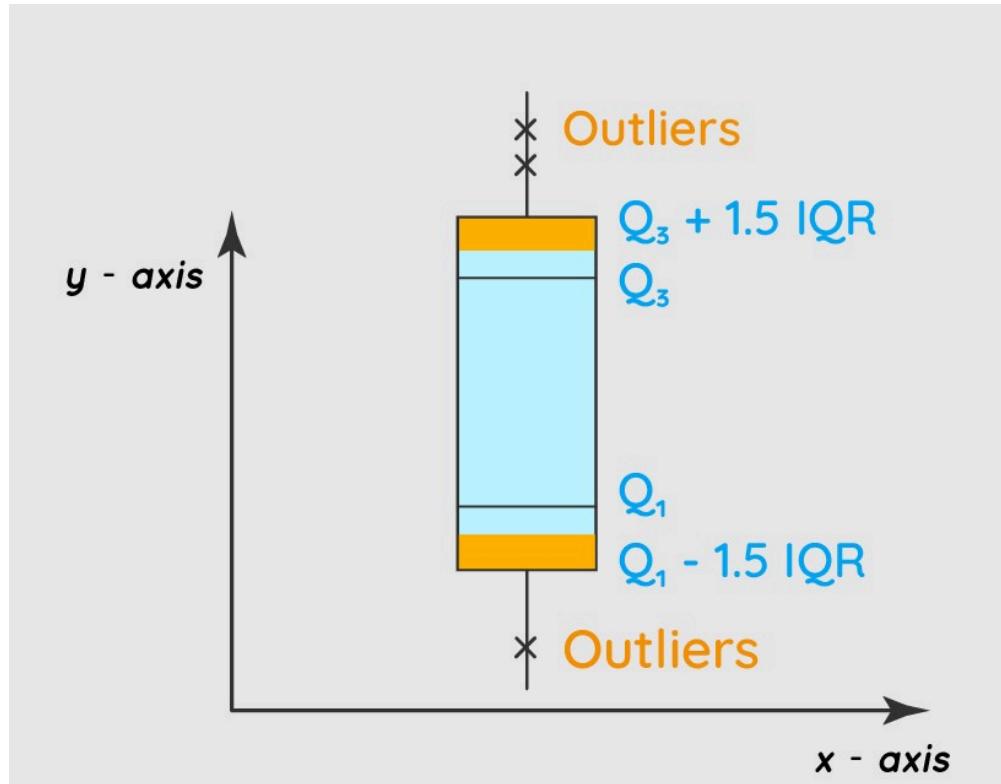
- Podemos ver que há valores discrepantes em algumas variáveis. Em credit_limit, income e reported_income possui os maiores valores discrepantes.

Outliers

- Um outlier é um ponto dos dados que é significativamente diferente dos dados restantes. É uma observação que se desvia tanto das outras observações a ponto de levantar suspeitas de que foi gerada por um mecanismo diferente.
- Outliers também são chamados de anormalidades, discordantes, desviantes ou anomalias na literatura de mineração de dados e estatística. Na maioria das aplicações, os dados são criados por um ou mais processos de geração, que podem refletir a atividade no sistema ou observações coletadas sobre entidades. Quando o processo de geração se comporta de forma incomum, isso resulta na criação de outliers. Portanto, um outlier geralmente contém informações úteis sobre características anormais dos sistemas e entidades que impactam o processo de

geração de dados. O reconhecimento de tais características incomuns fornece insights úteis específicos da aplicação.

Identificação de existência de outliers univariados



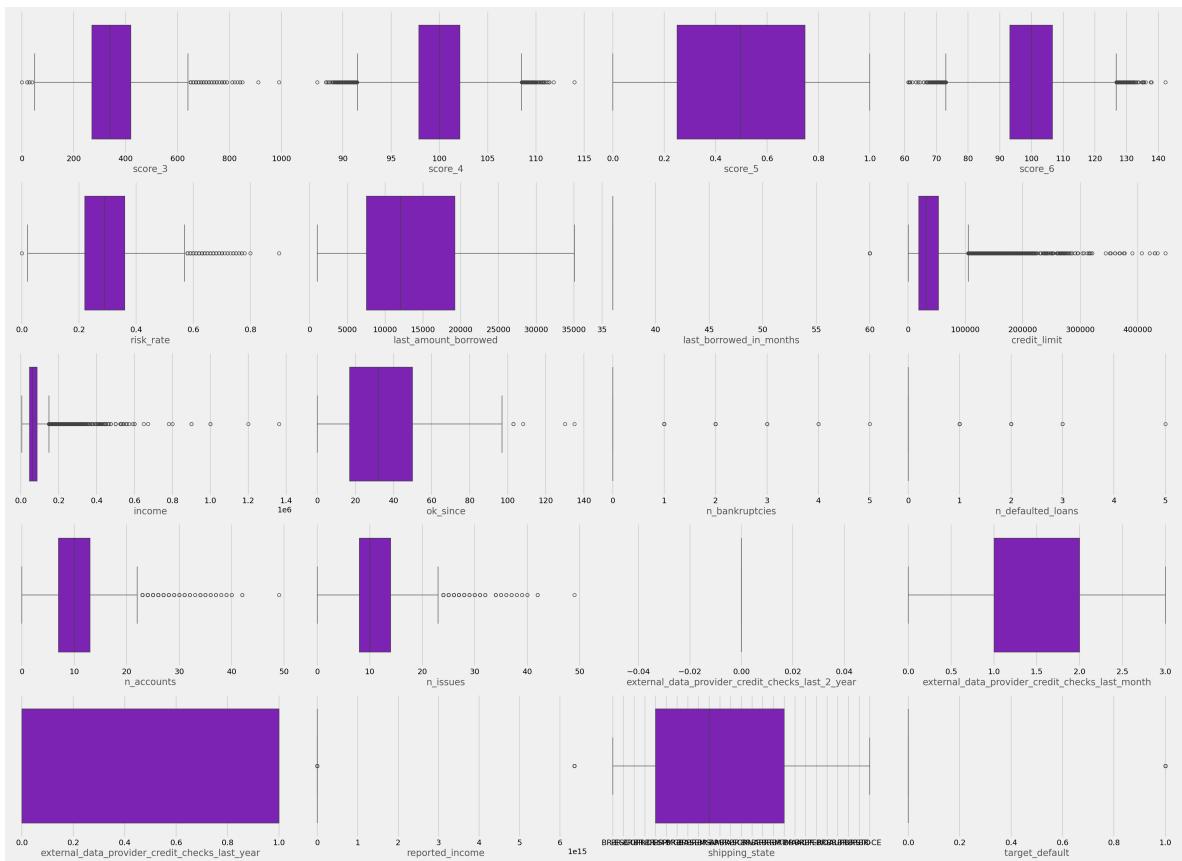
Este gráfico acima é o Boxplot.

O Boxplot, ou gráfico de caixa, é uma ferramenta estatística usada para resumir e visualizar a distribuição de um conjunto de dados. Ele fornece uma visão clara sobre a dispersão e as características principais dos dados, como mediana, quartis e possíveis valores atípicos.

Boxplot Dataframe Original

```
In [15]: plt.style.use("fivethirtyeight")
plt.figure(figsize=(30,30))
for index,column in enumerate(train):
    plt.subplot(7,4,index+1)
    sns.boxplot(data=train,x=column, color="#830BD1")

plt.tight_layout(pad = 1.0)
```



- Podemos ver que há Outliers praticamente em todas as variáveis numéricas.
- Devido a essa quantidade de valores discrepantes, neste primeiro CRISP-DM, vamos remover os Outliers.

```
In [16]: print("Dataframe Original:")
print(train.head())

train_no_outliers = detectar_e_substituir_outliers(train)

print("\Dataframe atualizado:")
print(train_no_outliers.head())
```

Dataframe Original:

	score_3	score_4	score_5	sco
re_6	risk_rate	last_amount_borrowed	last_borrowed_in_months	
credit_limit		income	ok_since	n_bankruptcies
defaulted_loans		n_accounts	n_issues	external_data_provider
_credit_checks_last_2_year		external_data_provider_credit_checks_last_month	exte	
rnal_data_provider_credit_checks_last_year		reported_income	shipping_state	ta
target_default				rget_default
0	240.00	105.33	0.89	9
4.50	0.26	NaN	NaN	
21,968.00	45,013.96	NaN	1.00	
0.00	11.00	11.00		
0.00		2		
NaN	92,586.00	BR-ES	False	
1	270.00	103.21	0.61	10
1.57	0.33	19,237.32	36.00	
40,972.00	80,022.23	NaN	0.00	
0.00	12.00	12.00		
0.00		3		
1.00	95,975.00	BR-GO	False	
2	260.00	102.56	0.92	11
7.06	0.28	NaN	NaN	
NaN	19,225.52	NaN	0.00	
0.00	7.00	NaN		
NaN		1		
NaN	53,981.00	BR-PR	False	
3	300.00	101.66	0.63	8
9.14	0.28	NaN	NaN	
NaN	60,043.78	62.00	NaN	
0.00	9.00	NaN		
0.00		2		
NaN	140,976.00	BR-DF	False	
4	350.00	109.60	0.66	9
8.57	0.27	NaN	NaN	
156,549.00	45,032.90	45.00	0.00	
0.00	25.00	25.00		
NaN		3		
NaN	120,129.00	BR-SP	False	

Os valores discrepantes foram substituídos por NaN.

\Dataframe atualizado:

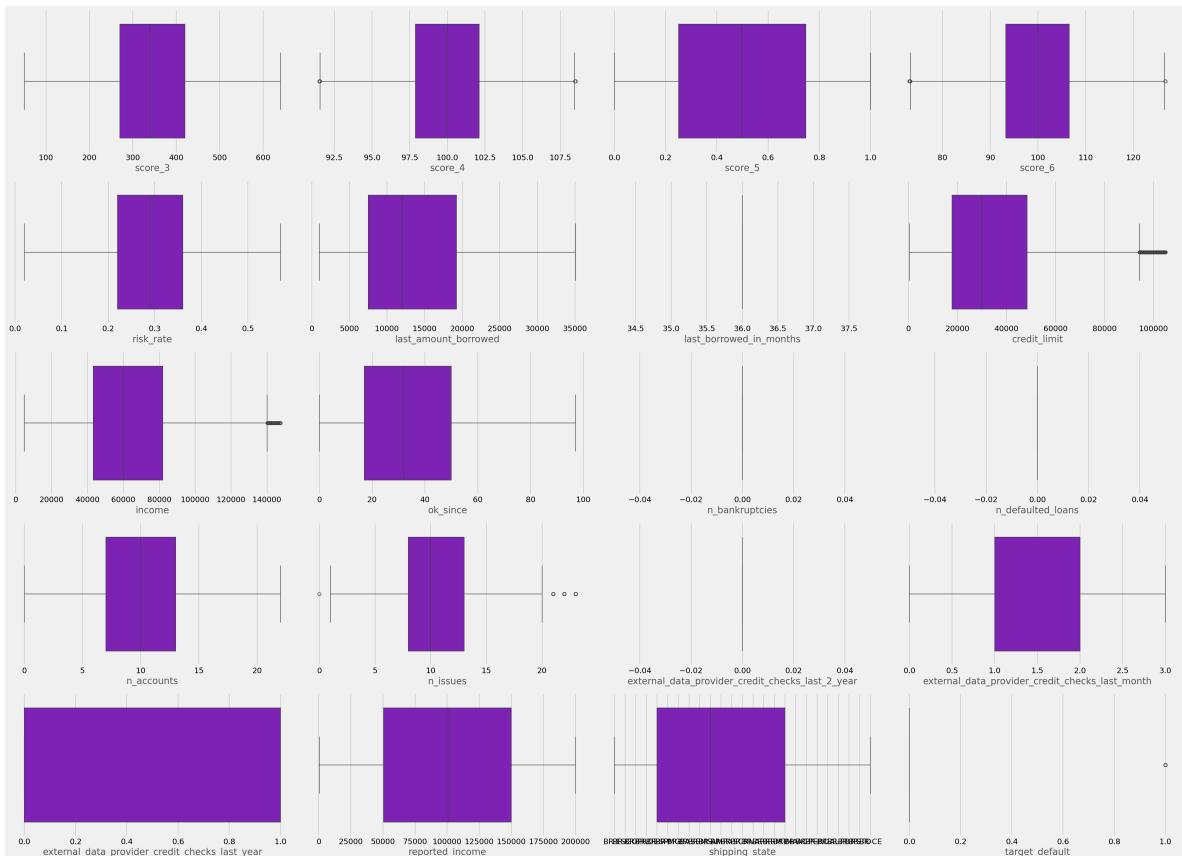
	score_3	score_4	score_5	sco
re_6	risk_rate	last_amount_borrowed	last_borrowed_in_months	
credit_limit		income	ok_since	n_bankruptcies
defaulted_loans		n_accounts	n_issues	external_data_provider
_credit_checks_last_2_year		external_data_provider_credit_checks_last_month	exte	
rnal_data_provider_credit_checks_last_year		reported_income	shipping_state	ta
target_default				rget_default
0	240.00	105.33	0.89	9
4.50	0.26	NaN	NaN	
21,968.00	45,013.96	NaN	NaN	
0.00	11.00	11.00		
0.00		2		
NaN	92,586.00	BR-ES	False	
1	270.00	103.21	0.61	10
1.57	0.33	19,237.32	36.00	
40,972.00	80,022.23	NaN	0.00	
0.00	12.00	12.00		
0.00		3		
1.00	95,975.00	BR-GO	False	
2	260.00	102.56	0.92	11

eda						
7.06	0.28	NaN	NaN	NaN	NaN	NaN
NaN	19,225.52	NaN	NaN	0.00	0.00	NaN
0.00	7.00	NaN	NaN	NaN	NaN	NaN
NaN	53,981.00	BR-PR	False	1	NaN	NaN
3	300.00	101.66	0.63	8	NaN	NaN
9.14	0.28	NaN	NaN	NaN	NaN	NaN
NaN	60,043.78	62.00	NaN	NaN	NaN	NaN
0.00	9.00	NaN	NaN	NaN	NaN	NaN
0.00	140,976.00	BR-DF	False	2	0.66	9
4	350.00	NaN	NaN	NaN	NaN	NaN
8.57	0.27	NaN	NaN	NaN	NaN	NaN
NaN	45,032.90	45.00	0.00	0.00	0.00	NaN
0.00	NaN	NaN	NaN	NaN	NaN	NaN
NaN	120,129.00	BR-SP	False	3	NaN	NaN

Boxplot Dataframe Atualizado

```
In [17]: plt.style.use("fivethirtyeight")
plt.figure(figsize=(30,30))
for index,column in enumerate(train_no_outliers):
    plt.subplot(7,4,index+1)
    sns.boxplot(data=train_no_outliers,x=column, color="#830BD1")

plt.tight_layout(pad = 1.0)
```



- Podemos observar que agora as variáveis numéricas que possuíam muitos valores discrepantes agora estão mais equilibrados.

Input Missing

```
In [18]: # SEPARAÇÃO DAS COLUNAS DAS VARIÁVEIS NUMÉRICAS E CATEGÓRICAS
train_no_outliers_num_columns = train_no_outliers.select_dtypes(exclude='object')
train_no_outliers_cat_columns = train_no_outliers.select_dtypes(include='object')

In [19]: # Input Missing Variáveis Numéricas

imputer = SimpleImputer(missing_values=np.nan, strategy='median')
imputer = imputer.fit(train_no_outliers.loc[:, train_no_outliers_num_columns])
train_no_outliers.loc[:, train_no_outliers_num_columns] = imputer.transform(train_no_outliers)

# Input Missing Variáveis Categóricas

imputer = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
imputer = imputer.fit(train_no_outliers.loc[:, train_no_outliers_cat_columns])
train_no_outliers.loc[:, train_no_outliers_cat_columns] = imputer.transform(train_no_outliers)
```

Alteração da Variável "target_default" para valores numéricos 0 e 1.

- Para que possamos utilizar as ferramentas de estatística e os modelos de machine learning, precisamos converter valores binários com True e False para 1 e 0, pois todos os cálculos são feitos com números.

```
In [20]: train_no_outliers['target_default'] = train_no_outliers['target_default'].astype
```

Separação das Variáveis Numéricas e Categóricas

```
In [21]: train_no_outliers_numerical = train_no_outliers.select_dtypes(exclude='object')
train_no_outliers_categorical = train_no_outliers.select_dtypes(include='object')

In [22]: train_no_outliers.isna().sum()
```

```
Out[22]: score_3          0
          score_4          0
          score_5          0
          score_6          0
          risk_rate         0
          last_amount_borrowed 0
          last_borrowed_in_months 0
          credit_limit        0
          income              0
          ok_since            0
          n_bankruptcies      0
          n_defaulted_loans    0
          n_accounts           0
          n_issues             0
          external_data_provider_credit_checks_last_2_year 0
          external_data_provider_credit_checks_last_month   0
          external_data_provider_credit_checks_last_year    0
          reported_income       0
          shipping_state        0
          target_default        0
          dtype: int64
```

Análise Univariada

Score

O score de crédito é geralmente calculado com base em modelos estatísticos que analisam o histórico financeiro de uma pessoa ou empresa. Esses modelos consideram uma variedade de informações, incluindo:

- Histórico de pagamento: Frequência com que você paga suas contas no prazo.
- Dívidas em aberto: Valor e número de dívidas existentes.
- Uso do crédito: Proporção de crédito usado em relação ao limite disponível (chamado de "taxa de utilização").
- Tempo de crédito: Quantidade de tempo que você possui histórico financeiro.
- Consultas de crédito: Quantas vezes seu relatório de crédito foi consultado recentemente.
- Registros negativos: Presença de atrasos, inadimplências, protestos ou ações judiciais.

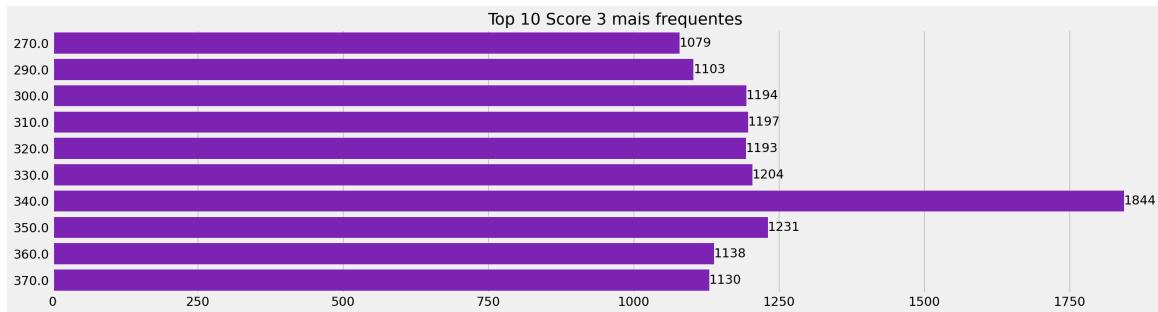
O score de crédito é tipicamente representado por um número dentro de um intervalo, como:

- 300 a 850 (nos EUA, em modelos como FICO ou VantageScore)
- 0 a 1.000 (no Brasil, por birôs como Serasa, Boa Vista ou SPC)

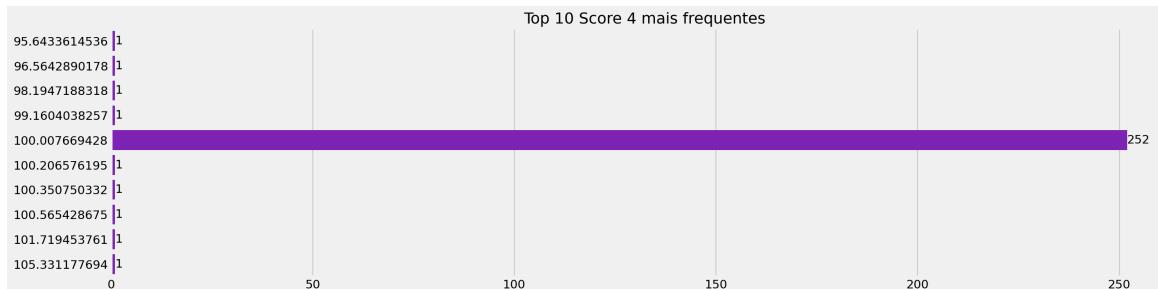
O score brasileiro é geralmente dividido em faixas:

- Baixo (0-300): Alto risco de inadimplência.
- Médio (301-700): Risco moderado.
- Alto (701-1000): Baixo risco de inadimplência.

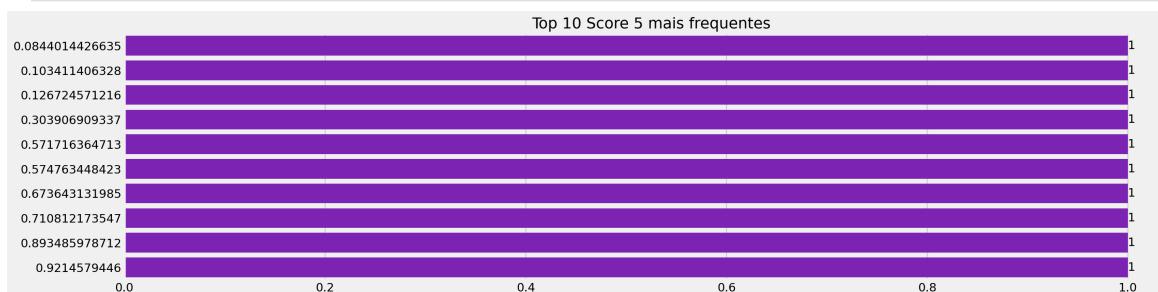
In [23]: #SCORE_3
top_10_score_plot(train_no_outliers, "score_3", "Top 10 Score 3 mais frequentes")



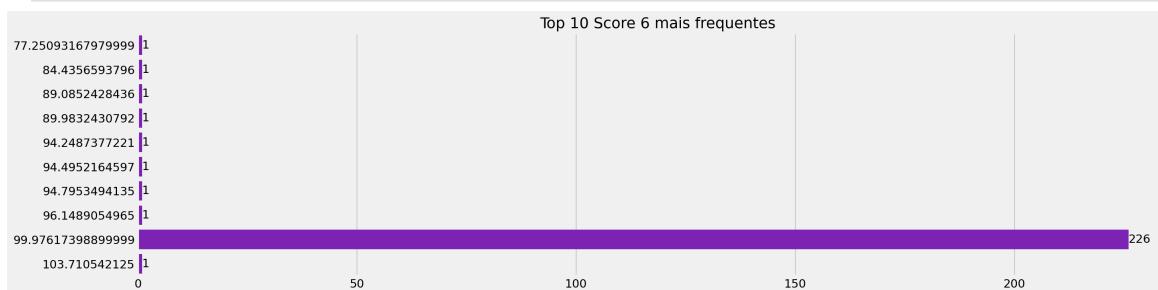
In [24]: #SCORE_4
top_10_score_plot(train_no_outliers, "score_4", "Top 10 Score 4 mais frequentes")



In [25]: #SCORE_5
top_10_score_plot(train_no_outliers, "score_5", "Top 10 Score 5 mais frequentes")



In [26]: #SCORE_6
top_10_score_plot(train_no_outliers, "score_6", "Top 10 Score 6 mais frequentes")



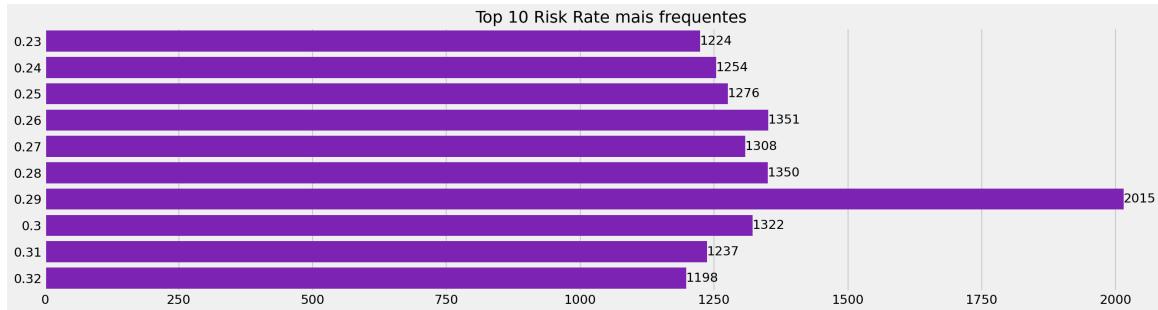
- Entre os Scores vou utilizar o score_3 por ser mais robusto para comparação com outras variáveis, possui uma distribuição de frequência melhor.

Risk Rate

O termo "risk rate" (taxa de risco) se refere à probabilidade ou ao nível de risco associado a uma determinada situação, investimento, crédito ou evento. Dependendo do

contexto, pode ser calculado de diferentes maneiras, mas, em geral, o "risk rate" indica o grau de risco envolvido e pode influenciar decisões financeiras, políticas ou operacionais.

In [27]: `#RISK RATE
top_10_score_plot(train_no_outliers, "risk_rate", "Top 10 Risk Rate mais frequen`



Risk Rate máximo para comparação.

In [28]: `train_no_outliers.risk_rate.max()`

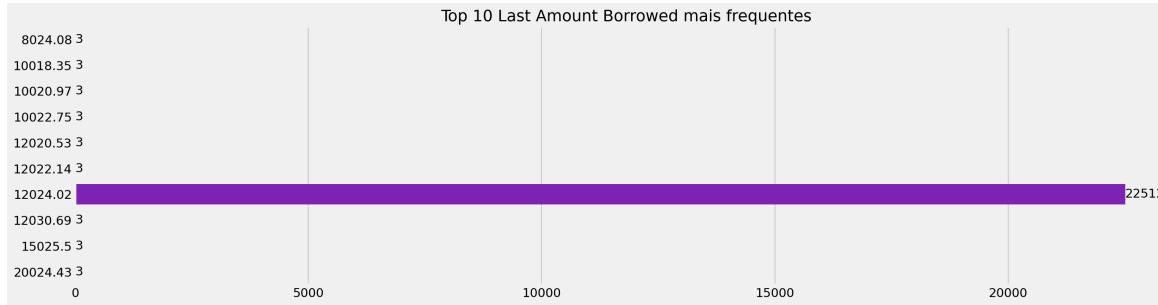
Out[28]: 0.57

- 0.29 é taxa de risk rate mais frequente, seguida de 0.28 e 0.26, onde o valor máximo de risk rate é 0.9.

Last Amount Borrowed

"Last Amount Borrowed" (em português, "Último Valor Emprestado") refere-se ao valor mais recente que você tomou emprestado em uma transação financeira, como um empréstimo bancário, um crédito rotativo ou qualquer outro tipo de operação de crédito. Esse termo geralmente é usado em relatórios financeiros, aplicativos de empréstimos ou plataformas de gestão financeira para indicar o valor do empréstimo mais recente.

In [29]: `#Last Amount Borrowed
top_10_score_plot(train_no_outliers, "last_amount_borrowed", "Top 10 Last Amount`

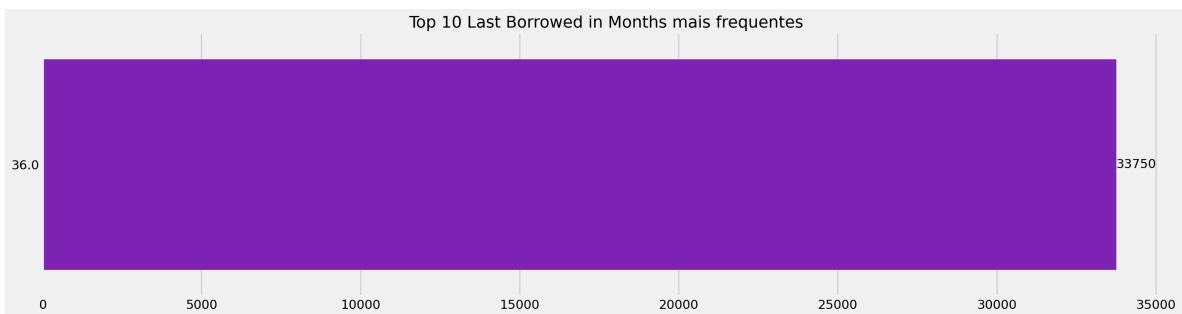


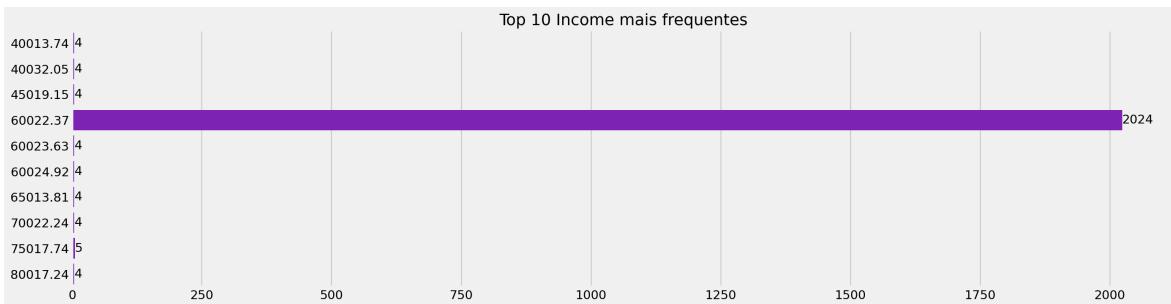
- Temos algo interessante, o valor de 12024.02 é praticamente o valor mais frequente de todos.

Last Borrowed in Months

"Last Borrowed in Months" (em português, "Último Empréstimo em Meses") refere-se ao número de meses que se passaram desde a última vez que você tomou um empréstimo ou utilizou crédito.

```
In [30]: #Last Borrowed in Months
top_10_score_plot(train_no_outliers, "last_borrowed_in_months", "Top 10 Last Bor
```





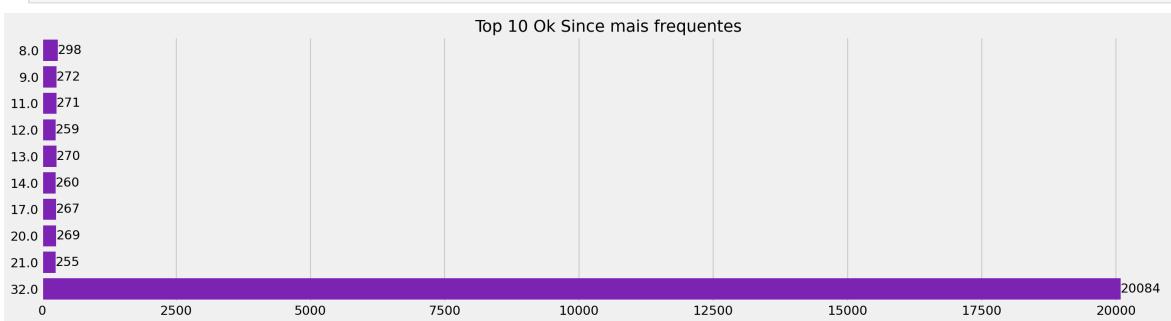
- E outra vez temos algo interessante, disparado com maior frequência temos o valor de 60,022.37 com o valor de renda.

Ok since

"Ok Since" (em português, algo como "Está Ok Desde") é um termo geralmente usado em relatórios financeiros, de crédito ou em sistemas de análise de dados para indicar a data ou o período em que um registro ou situação começou a ser considerado positivo, estável ou em conformidade.

- O "OK Since" é frequentemente usado em relatórios de crédito para indicar desde quando uma conta está ativa sem incidentes, como atrasos de pagamento ou inadimplência.
- Uma conta com um "OK Since" antigo pode ser vista como mais estável e confiável, pois demonstra um histórico mais longo de bom comportamento de crédito.
- Por outro lado, um "OK Since" recente pode sugerir que a conta estava anteriormente em situação de risco ou irregularidade.

```
In [33]: #OK SINCE
top_10_score_plot(train_no_outliers, "ok_since", "Top 10 Ok Since mais frequente")
```

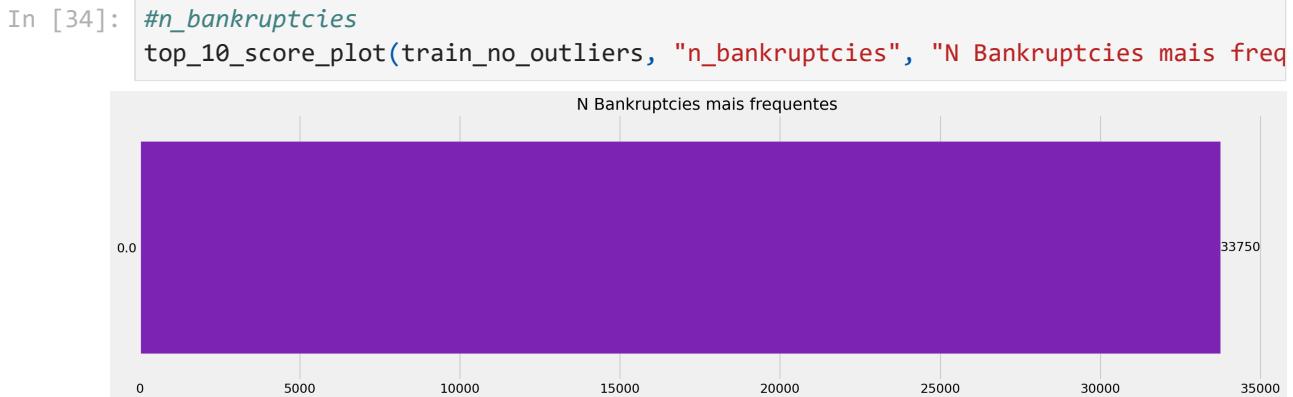


- Aqui mais outro valor interessante, o valor mais frequente é o valor com 32.0, provavelmente esses valores significam 32 semanas, ou 32 meses, ou 32 dias pra indicar que um empréstimo está em bom estado, sem atrasos ou problemas.

N Bankruptcies

"N Bankruptcies" (em português, "Número de Falências" ou "Número de Insolvências") refere-se à quantidade de vezes que uma pessoa ou empresa declarou falência ou

insolvência oficialmente. É uma métrica usada, principalmente, em relatórios de crédito e análise financeira, para avaliar o histórico financeiro de um indivíduo ou organização.



- Podemos ver que os clientes não tiveram declaração de falência.

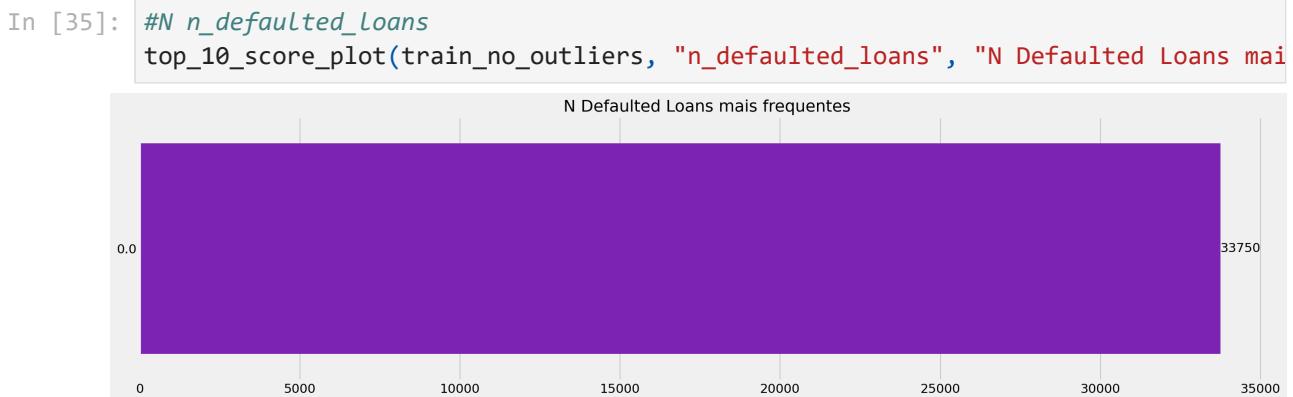
N Defaulted Loans

"N Defaulted Loans" (em português, "Número de Empréstimos em Inadimplência") refere-se à quantidade de empréstimos que uma pessoa ou empresa deixou de pagar de acordo com os termos estabelecidos no contrato, resultando em inadimplência. Essa métrica é usada principalmente em relatórios de crédito e análises financeiras para avaliar o histórico de pagamentos e a confiabilidade de um tomador de crédito.

Exemplo de **N Defaulted Loans**:

N = 0: Nunca teve um empréstimo em inadimplência.

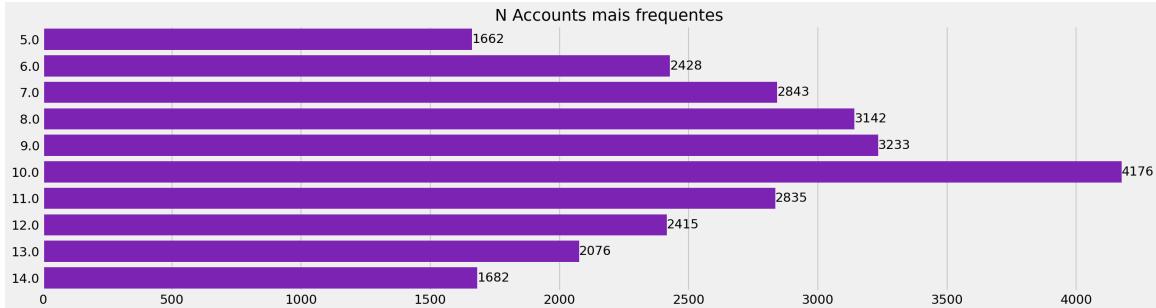
N = 3: Teve três empréstimos em inadimplência.



- Podemos ver que número de empréstimos não possuem inadimplência.

N Accounts

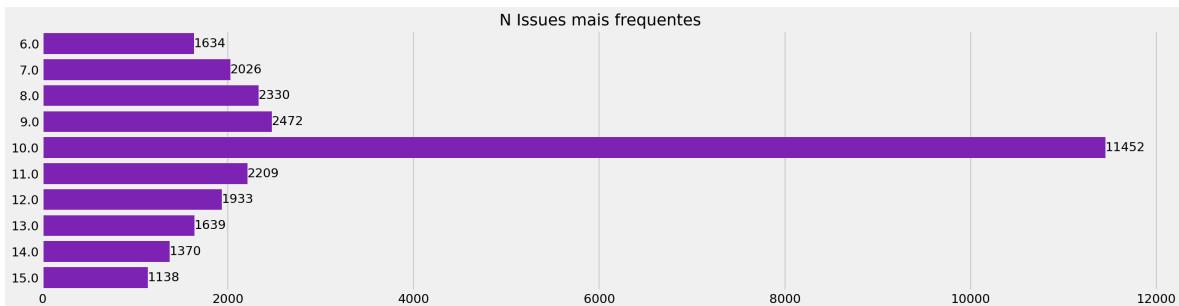
"N Accounts" (em português, "Número de Contas") refere-se à quantidade total de contas financeiras ou de crédito que uma pessoa ou empresa possui em seu nome. Esse termo geralmente aparece em relatórios de crédito, avaliações financeiras ou sistemas de gerenciamento financeiro.

In [36]: `#n_accounts``top_10_score_plot(train_no_outliers, "n_accounts", "N Accounts mais frequentes")`

- Podemos ver que 4,176 clientes possuem pelo menos 10 contas financeiras em seus nomes, seguido de 3,233 clientes com 9 contas e 3,142 clientes com 8 contas financeiras.

N Issues

"N Issues" (em português, "Número de Problemas" ou "Número de Ocorrências") refere-se à quantidade de incidentes, irregularidades ou situações negativas associadas a uma conta, perfil financeiro ou histórico de crédito. Esse termo é frequentemente usado em relatórios financeiros ou sistemas de análise para destacar possíveis preocupações relacionadas ao comportamento financeiro de uma pessoa ou empresa.

In [37]: `#n_accounts``top_10_score_plot(train_no_outliers, "n_issues", "N Issues mais frequentes")`

- Podemos ver que 10 Números de Problemas associadas as contas dos clientes são maioria.

External Data Provider Credit Checks Last 2 Years

"External Data Provider Credit Checks Last 2 Years" (em português, "Consultas de Crédito por Provedores Externos nos Últimos 2 Anos") refere-se ao número de vezes que provedores externos de dados financeiros ou instituições, como bancos ou empresas de crédito, consultaram o histórico de crédito de uma pessoa ou empresa nos últimos dois anos.

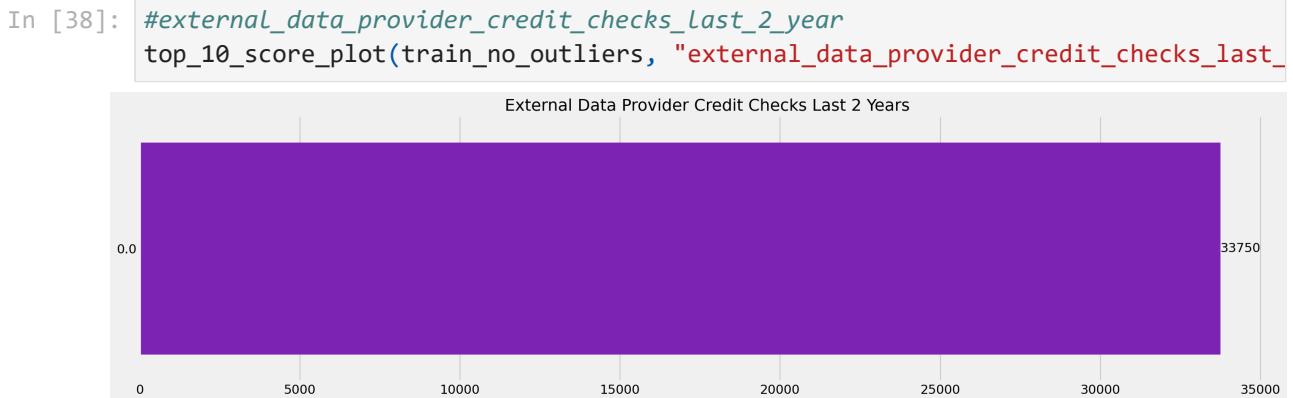
O que são Provedores Externos?

- São entidades que acessam informações financeiras para avaliar o risco de crédito, como birôs de crédito (por exemplo, Serasa, SPC, ou Boa Vista no Brasil).

- Essas consultas são realizadas por credores ou empresas ao analisar uma solicitação de crédito, empréstimos, cartões de crédito ou financiamentos.

Exemplos de "Credit Checks Last 2 Years":

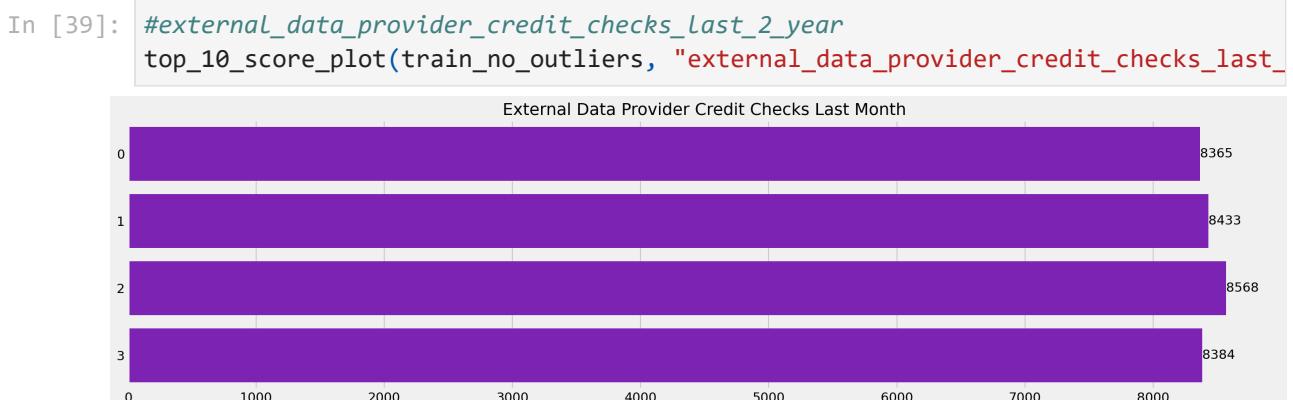
- N = 3: Significa que seu histórico de crédito foi consultado 3 vezes por provedores externos nos últimos dois anos.
- N = 10: Dez consultas foram realizadas, indicando maior atividade ou interesse em suas informações financeiras.



- Podemos ver que 0 foi a quantidade de vezes que os provedores externos consultaram o histórico de crédito.

External Data Provider Credit Checks Last Month

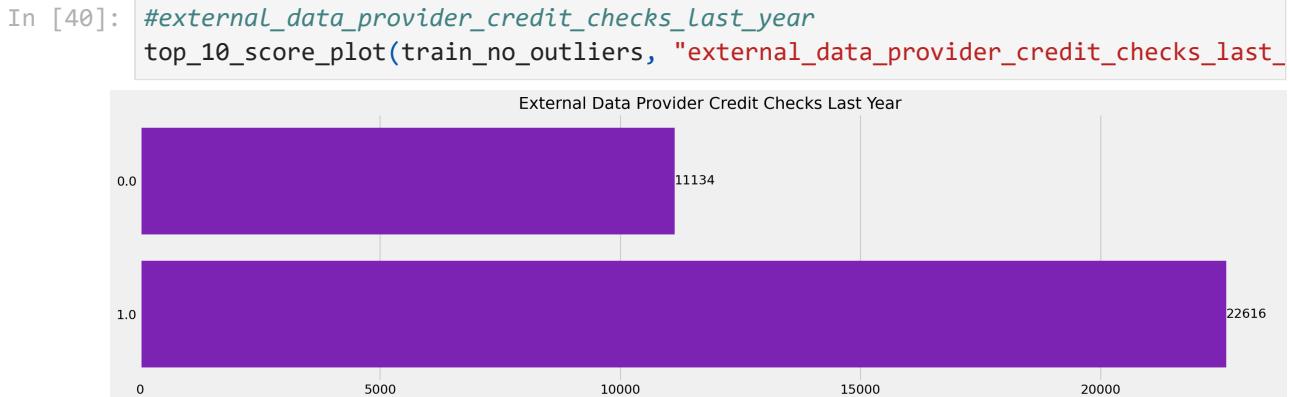
"External Data Provider Credit Checks Last Month" (em português, "Consultas de Crédito por Provedores Externos no Último Mês") refere-se ao número de vezes que provedores externos de dados financeiros, como birôs de crédito (por exemplo, Serasa, SPC ou Boa Vista no Brasil), foram acionados para verificar seu histórico de crédito no mês anterior.



- Podemos ver que 2 foi a quantidade de vezes que um provedor externo consultou o histórico de crédito, seguido de 1, depois 3 vezes.

External Data Provider Credit Checks Last Year

"External Data Provider Credit Checks Last Year" (em português, "Consultas de Crédito por Provedores Externos no Último Ano") refere-se ao número de vezes que instituições financeiras, empresas ou terceiros consultaram seu histórico de crédito em provedores externos de dados financeiros, como birôs de crédito (Serasa, SPC, Boa Vista, entre outros), ao longo do último ano.



- Podemos ver que 1 vez foi a quantidade de vezes que o histórico dos clientes foram consultados com maior frequência.

Reported Income

"Reported Income" (em português, "Renda Declarada") refere-se ao valor de renda que uma pessoa ou empresa informa oficialmente a uma instituição ou órgão, geralmente em situações como:

1. Solicitações de Crédito:

É o valor da renda informado pelo cliente ao solicitar um empréstimo, financiamento ou cartão de crédito. Pode incluir salários, lucros, aluguéis, pensões ou outras fontes de receita.

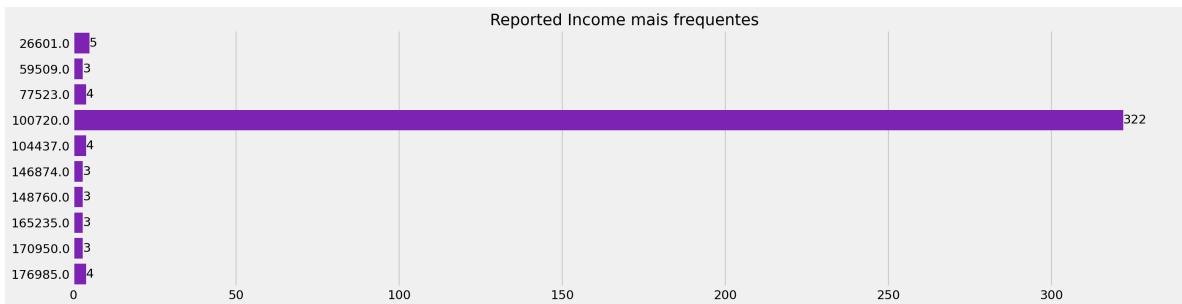
2. Declarações Fiscais:

É o valor registrado em documentos oficiais, como declarações de imposto de renda, para fins tributários. Esse valor deve ser compatível com os rendimentos reais para evitar penalidades legais.

3. Verificações de Renda:

Algumas instituições podem solicitar comprovantes, como contracheques, extratos bancários ou declarações fiscais, para validar o "reported income".



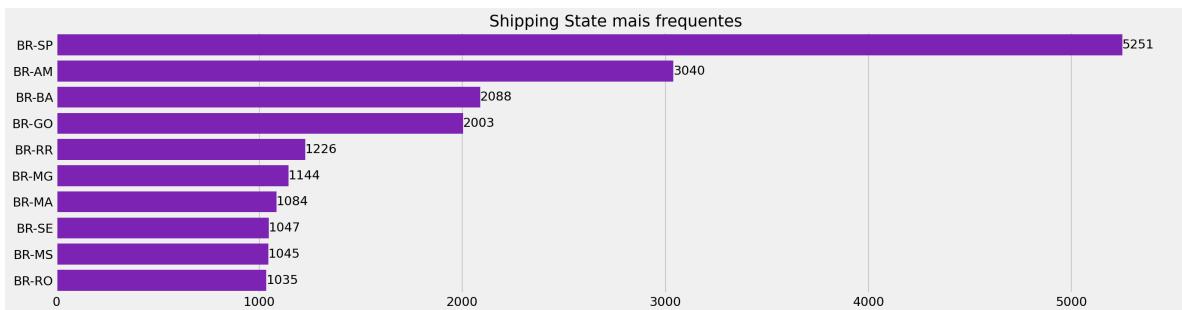


- O valor de 100,720,00 é o mais frequente.
- Em segundo lugar vem o valor de 26,601,00.

Shipping State

"Shipping State" (em português, "Estado de Envio") refere-se ao estado (ou província, dependendo do país) onde um pacote ou produto será enviado ou está sendo enviado. Esse termo é comumente usado em processos de compras online e logística para identificar a localização de entrega dentro de um país.

```
In [42]: #shipping_state
top_10_score_plot(train_no_outliers, "shipping_state", "Shipping State mais freq
```



- Podemos ver que SP é o estado com mais frequência, em segundo vem o AM.

Target Default

Target default no contexto de credit risk (risco de crédito) refere-se à definição de inadimplência que é usada como a variável alvo (target) em modelos de previsão de risco de crédito. Essa variável indica se um cliente (pessoa física ou jurídica) cumpriu ou não suas obrigações financeiras, como pagamentos de empréstimos ou financiamentos.

```
In [43]: #Target Default
top_10_score_plot(train_no_outliers, "target_default", "Target Default mais freq
```



- Há 28.680 clientes sem risco de crédito contra 5.070 com risco de crédito.

Relatório Análise Univariada

- Entre os **Scores**, o Score 3 é o mais diversificado para uma análise bi e multivariada, onde o score **340.0** é o valor mais frequente.
- Em **Risk Rate** o valor de **0.29** é o valor mais frequente, onde o valor máximo registrado é de **0.57**.
- Em **Last Amount Borrowed** o valor de **12,024.02** é o mais frequente, e praticamente isolado, pois em segundo lugar vem o valor de 10022.75 com apenas 3 registros.
- Em **Last Borrowed in Months** **36,0** meses é o intervalo com mais frequência.
- Em **Credit Limit** o valor de **29,942.00** é o valor mais frequente, e praticamente isolado, pois em segundo lugar vem o valor de **10,000.0** com apenas **107** registros.
- Em **Income** o valor de **60,022.37** é o mais frequente, e praticamente isolado, pois em segundo lugar vem o valor de **75,017.74** com apenas 5 registros.
- Em **Ok Since** temos o valor de **32,0** com mais frequência e praticamente isolado, com **20,084** registros.
- Em **N Bankruptcies** podemos ver que os clientes não tiveram declaração de falência.
- Em **N Defaulted Loans** podemos ver que número de empréstimos não possuem inadimplência.
- Em **N Accounts** podemos ver que grande parte dos clientes possuem mais de **5** a **10** contas ou créditos em seus nomes.
- Em **N Issues** podemos ver que **10** problemas ou ocorrências, irregularidades associadas as contas dos clientes são mais frequentes, seguidas de 9, 8 e 11 ocorrências.
- Em **External Data Provider Credit Checks Last 2 Years** possuem **0** quantidade de consultas.
- Em **External Data Provider Credit Checks Last Month** já é um pouco diferente. **2** é o números mais frequente de vezes que provedores externos consultaram o histórico de crédito no mês anterior.
- Em **External Data Provider Credit Checks Last Year** **1** que significa True é o numero com mais frequência em consultas dos dados históricos de crédito.

- Em **Reported Income** o valor de 100,720.00 é o mais frequente. Em segundo lugar vem o valor de 26,601.00.
- Em **Shipping State** é basicamente o registro dos estados onde se localizam cada cliente. **SP** é o estado com mais frequência.
- E em **Target Default** podemos ver que há mais registros **positivos** com relação aos créditos.

Análise Bivariada

A análise bivariada é uma técnica estatística utilizada para examinar a relação entre duas variáveis, com o objetivo de identificar associações, padrões ou dependências entre elas. Essa análise é fundamental para entender como uma variável pode influenciar ou estar relacionada com outra.

A análise bivariada pode ser estudada a partir das seguintes situações:

1. Variáveis categóricas:

- Exemplo: Relação entre gênero (masculino/feminino) e preferência por um produto. Método comum: Tabelas de contingência (tabelas cruzadas) e o teste qui-quadrado.

2. Uma variável categórica e outra numérica:

- Exemplo: Relação entre nível educacional (fundamental, médio, superior) e renda mensal. Método comum: Testes de diferença de médias, como ANOVA ou teste t.

3. Duas variáveis numéricas:

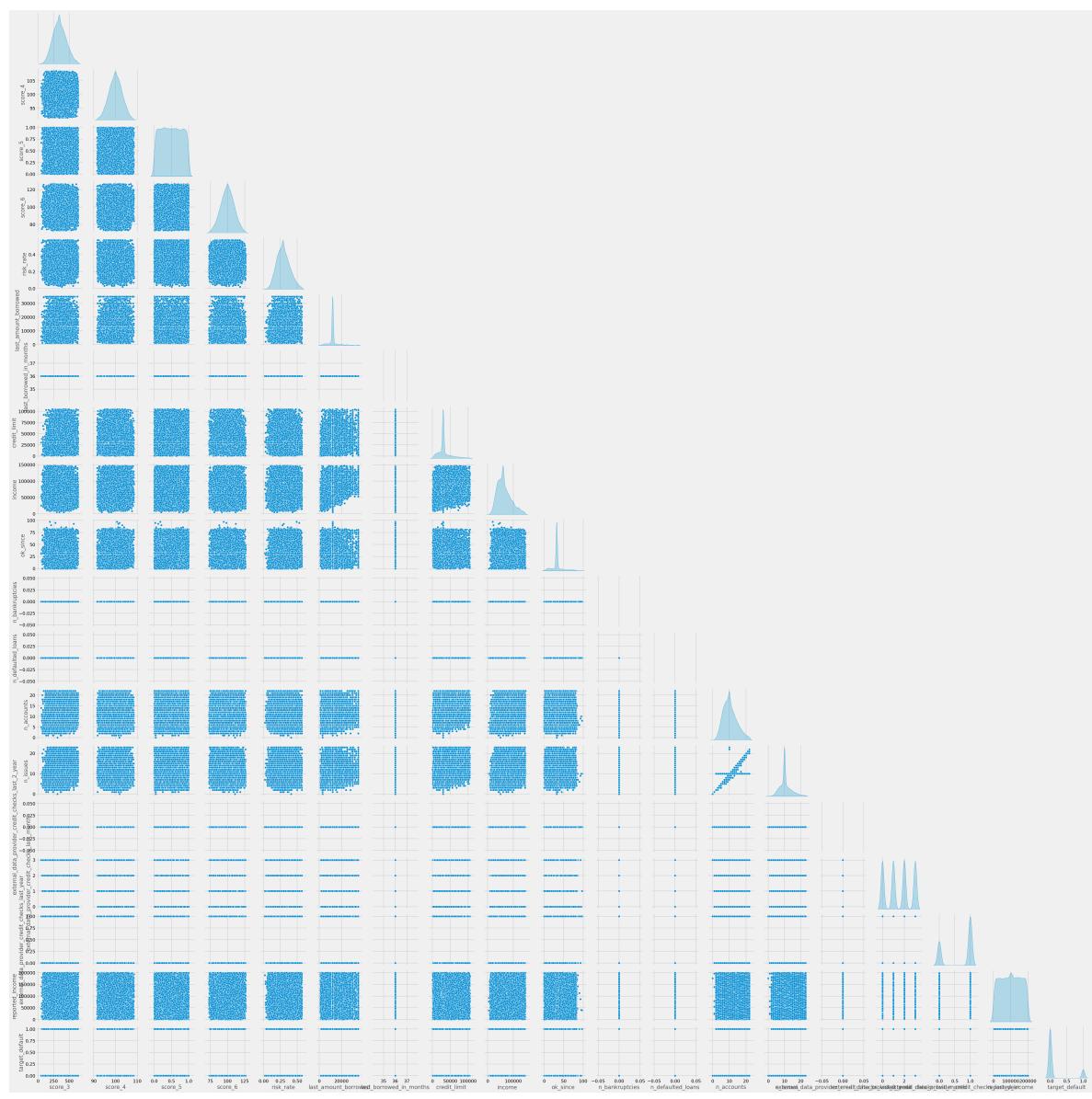
- Exemplo: Relação entre altura e peso de indivíduos. Método comum: Correlação (como o coeficiente de Pearson) e análise de regressão.

No nosso caso estamos comparando basicamente variáveis numéricas com a variável binária `target_default`. `shipping_state` não tem necessidade de comparação.

Diagramas de Dispersão

A correlação entre duas variáveis quantitativas pode ser representada de forma gráfica por meio de um diagrama de dispersão. Ele representa graficamente os valores das variáveis X e Y em um plano cartesiano.

```
In [44]: sns.pairplot(train_no_outliers_numerical, diag_kind="kde", corner=True, palette
plt.show()
```



- Podemos ver que não há uma correlação visível entre as variáveis.

Matriz de Correlação de Pearson

O coeficiente de correlação de Pearson é uma medida que varia entre -1 e 1. Por meio do sinal, é possível verificar o tipo de relação linear entre as duas variáveis analisadas (direção em que a variável Y aumenta ou diminui em função da variação de X).

Quanto mais próximo dos valores extremos, mais forte é a correlação entre elas:

- 1: Correlação positiva perfeita (quando uma variável aumenta, a outra também aumenta proporcionalmente).
- 1: Correlação negativa perfeita (quando uma variável aumenta, a outra diminui proporcionalmente).
- 0: Nenhuma correlação linear (as variáveis não têm relação linear, mas podem estar relacionadas de outra forma).

Matriz de Correlação nas variáveis numéricas com relação a variável target

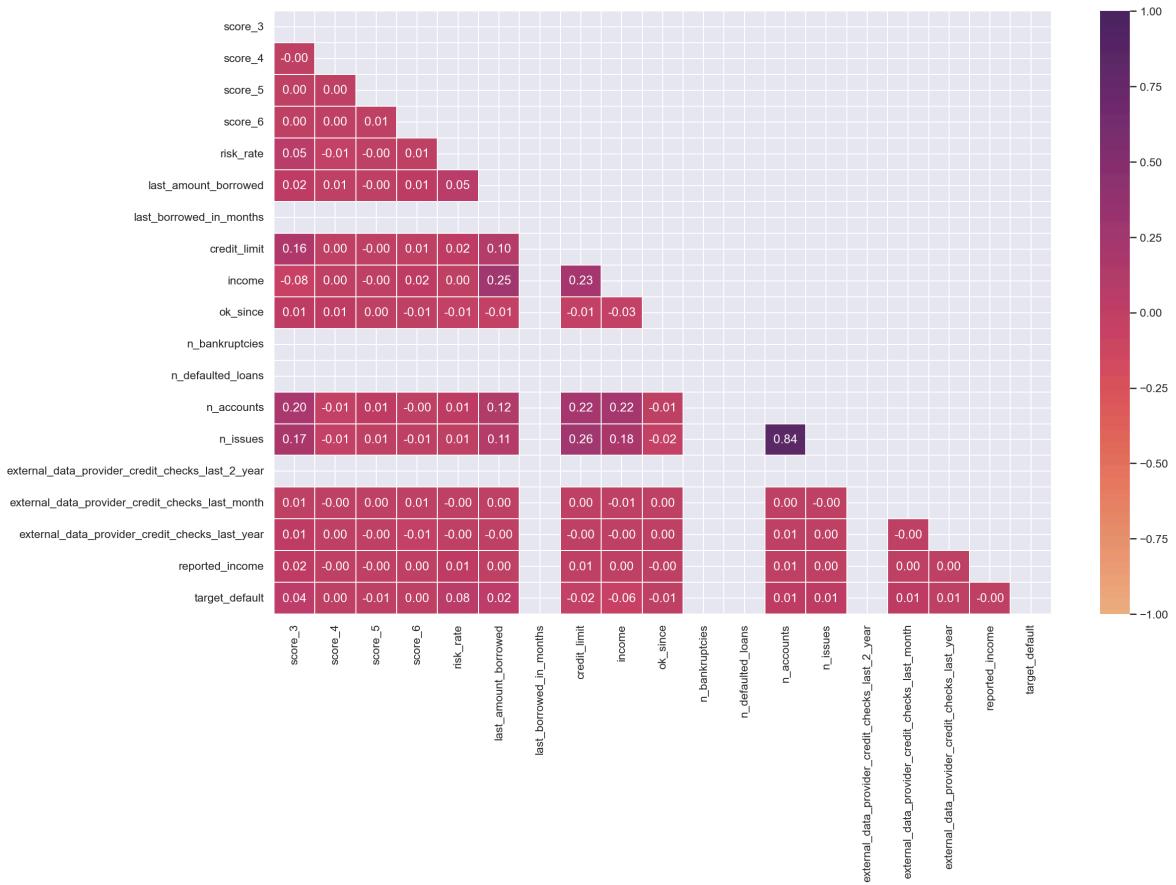
```
In [45]: corr_matrix = train_no_outliers.corr(numeric_only=True)
corr_matrix['target_default'].sort_values(ascending=False)
```

```
Out[45]: target_default          1.00
risk_rate                         0.08
score_3                           0.04
last_amount_borrowed              0.02
n_issues                          0.01
n_accounts                        0.01
external_data_provider_credit_checks_last_month 0.01
external_data_provider_credit_checks_last_year   0.01
score_4                           0.00
score_6                           0.00
reported_income                   -0.00
score_5                           -0.01
ok_since                          -0.01
credit_limit                      -0.02
income                            -0.06
last_borrowed_in_months           NaN
n_bankruptcies                   NaN
n_defaulted_loans                NaN
external_data_provider_credit_checks_last_2_year  NaN
Name: target_default, dtype: float64
```

- Podemos ver que não há uma correlação significativa nem com a variável target.

Heatmap da Correlação de Pearson em todas as variáveis numéricas

```
In [46]: sns.set(font_scale=1)
matrix = np.triu(train_no_outliers_numerical.corr())
plt.figure(figsize=(15, 10))
sns.heatmap(train_no_outliers_numerical.corr(), annot = True,
            cmap = 'flare', fmt=".2f",
            mask = matrix, vmin = -1, vmax = 1,
            linewidths = 0.1, linecolor = 'white')
plt.show()
```

**Tabela filtrada apenas com as correlações**

```
In [47]: corr_matrix = train_no_outliers_numerical.corr()

correlation_list = (
    corr_matrix
    .stack()
    .reset_index()
)
correlation_list.columns = ['Var1', 'Var2', 'Correlation']

unique_correlation_list = correlation_list[
    (correlation_list['Var1'] != correlation_list['Var2']) &
    (correlation_list['Var1'] < correlation_list['Var2'])
]

print(unique_correlation_list.sort_values(by='Correlation', ascending=False))
```

Var2	Correlation	Var1
145		n_accounts
n_issues	0.84	
100		credit_limit
n_issues	0.26	
110		income
last_amount_borrowed		0.25
97		credit_limit
income	0.23	
99		credit_limit
n_accounts	0.22	
114		income
n_accounts	0.22	
135		n_accounts
score_3	0.20	
115		income
n_issues	0.18	
150		n_issues
score_3	0.17	
90		credit_limit
score_3	0.16	
84		last_amount_borrowed
n_accounts	0.12	
85		last_amount_borrowed
n_issues	0.11	
95		credit_limit
last_amount_borrowed		0.10
74		risk_rate
target_default	0.08	
79		last_amount_borrowed
risk_rate	0.05	
60		risk_rate
score_3	0.05	
14		score_3
target_default	0.04	
89		last_amount_borrowed
target_default	0.02	
195		reported_income
score_3	0.02	
75		last_amount_borrowed
score_3	0.02	
94		credit_limit
risk_rate	0.02	
108		income
score_6	0.02	
164		n_issues
target_default	0.01	
137		n_accounts
score_5	0.01	
120		ok_since
score_3	0.01	
139		n_accounts
risk_rate	0.01	
180 external_data_provider_credit_checks_last_year		
score_3	0.01	
78		last_amount_borrowed
score_6	0.01	
149		n_accounts
target_default	0.01	

```

179 external_data_provider_credit_checks_last_month
target_default          0.01
121                      ok_since
score_4                 0.01
199                      reported_income
risk_rate                0.01
152                      n_issues
score_5                 0.01
148                      n_accounts
reported_income          0.01
194 external_data_provider_credit_checks_last_year
target_default          0.01
165 external_data_provider_credit_checks_last_month
score_3                 0.01
103                      credit_limit
reported_income          0.01
76                      last_amount_borrowed
score_4                 0.01
154                      n_issues
risk_rate                0.01
33                      score_5
score_6                 0.01
168 external_data_provider_credit_checks_last_month
score_6                 0.01
93                      credit_limit
score_6                 0.01
189 external_data_provider_credit_checks_last_year
n_accounts              0.01
63                      risk_rate
score_6                 0.01
91                      credit_limit
score_4                 0.00
2                      score_3
score_5                 0.00
188 external_data_provider_credit_checks_last_year
ok_since                0.00
181 external_data_provider_credit_checks_last_year
score_4                 0.00
106                      income
score_4                 0.00
190 external_data_provider_credit_checks_last_year
n_issues                0.00
163                      n_issues
reported_income          0.00
193 external_data_provider_credit_checks_last_year
reported_income          0.00
88                      last_amount_borrowed
reported_income          0.00
174 external_data_provider_credit_checks_last_month
n_accounts              0.00
178 external_data_provider_credit_checks_last_month
reported_income          0.00
101                      credit_limit  external_data_provider_cred
it_checks_last_month    0.00
29                      score_4
target_default          0.00
118                      income
reported_income          0.00
173 external_data_provider_credit_checks_last_month
ok_since                0.00

```

```

122                               ok_since
score_5                  0.00
170 external_data_provider_credit_checks_last_month
last_amount_borrowed      0.00
17                               score_4
score_5                  0.00
3                               score_3
score_6                  0.00
198                               reported_income
score_6                  0.00
167 external_data_provider_credit_checks_last_month
score_5                  0.00
109                               income
risk_rate                 0.00
59                               score_6
target_default             0.00
18                               score_4
score_6                  0.00
184 external_data_provider_credit_checks_last_year
risk_rate                 -0.00
166 external_data_provider_credit_checks_last_month
score_4                  -0.00
133                               ok_since
reported_income            -0.00
169 external_data_provider_credit_checks_last_month
risk_rate                 -0.00
197                               reported_income
score_5                  -0.00
175 external_data_provider_credit_checks_last_month
n_issues                 -0.00
102                               credit_limit    external_data_provider_cre
dit_checks_last_year       -0.00
92                               credit_limit
score_5                  -0.00
177 external_data_provider_credit_checks_last_month
dit_checks_last_year       -0.00
138                               n_accounts
score_6                  -0.00
182 external_data_provider_credit_checks_last_year
score_5                  -0.00
209                               reported_income
target_default             -0.00
62                               risk_rate
score_5                  -0.00
187 external_data_provider_credit_checks_last_year
income                  -0.00
77                               last_amount_borrowed
score_5                  -0.00
1                               score_3
score_4                  -0.00
185 external_data_provider_credit_checks_last_year
last_amount_borrowed      -0.00
107                               income
score_5                  -0.00
196                               reported_income
score_4                  -0.00
44                               score_5
target_default             -0.01
153                               n_issues
score_6                  -0.01

```

```

172 external_data_provider_credit_checks_last_month
income           -0.01
123                      ok_since
score_6          -0.01
143                      n_accounts
ok_since         -0.01
134                      ok_since
target_default    -0.01
61                      risk_rate
score_4          -0.01
183 external_data_provider_credit_checks_last_year
score_6          -0.01
136                      n_accounts
score_4          -0.01
151                      n_issues
score_4          -0.01
98                      credit_limit
ok_since         -0.01
83                      last_amount_borrowed
ok_since         -0.01
124                      ok_since
risk_rate        -0.01
158                      n_issues
ok_since         -0.02
104                      credit_limit
target_default   -0.02
113                      income
ok_since         -0.03
119                      income
target_default   -0.06
105                      income
score_3          -0.08

```

A força da correlação de Pearson depende do valor absoluto do coeficiente (r), que varia de -1 a 1. Embora não exista uma regra universal, existem diretrizes comuns amplamente aceitas:

- Uma correlação forte geralmente é considerada quando o valor absoluto de r é maior que 0.5.
- Uma correlação muito forte ocorre quando r é maior que 0.7.

Com isso podemos ver que não há uma correlação significativa entre as variáveis, apenas entre n_issues e $n_accounts$.

Considerações importantes

1. Correlação não implica causalidade:

- Mesmo uma correlação forte não significa que uma variável causa a outra.

2. Outliers podem distorcer r - o Coeficiente de Pearson:

- Valores extremos podem aumentar ou diminuir a correlação, tornando-a enganosa.

3. Correlação Linear:

- Pearson mede apenas relações lineares. Relações não lineares podem ter $r \approx 0$, mesmo que exista uma associação significativa entre as variáveis.

Multicolinearidade

Multicolinearidade é uma condição em que duas ou mais variáveis independentes em um modelo de regressão estão fortemente correlacionadas. Isso significa que as variáveis fornecem informações redundantes sobre o que estão tentando prever, dificultando a análise da influência individual de cada variável no modelo.

O problema da Multicolinearidade ocorre quando há correlações muito elevadas entre variáveis explicativas e, em casos extremos, tais correlações podem ser perfeitas, indicando uma relação linear entre as variáveis.

Uma das principais causas da multicolinearidade é a existência de variáveis que apresentam a mesma tendência durante alguns períodos.

Para identificar a Multicolinearidade podemos usar o Coeficiente de Correlação de Pearson, mas há também o VIF (Variance Inflation Factor).

O VIF mede o quanto a variância do coeficiente de uma variável aumenta devido à multicolinearidade. Valores acima de 5 (ou 10, dependendo do contexto) indicam problemas.

Interpretação dos Valores de VIF

VIF=1 -> Sem colinearidade

1 < VIF ≤ 5 -> Colinearidade moderada (normalmente aceitável)

VIF > 5 -> Colinearidade alta (potencial problema, investigar)

VIF > 10 -> Colinearidade severa (geralmente inaceitável, ajustar o modelo)

```
In [48]: X = train_no_outliers[['score_3', 'score_4', 'score_5', 'score_6', 'risk_rate',
   'last_borrowed_in_months', 'credit_limit', 'income', 'ok_since', 'n_b
   'n_defaulted_loans', 'n_accounts', 'n_issues', 'external_data_provide
   'external_data_provider_credit_checks_last_month',
   'external_data_provider_credit_checks_last_year', 'reported_income']]

vif_data = pd.DataFrame()
vif_data["Feature"] = X.columns

vif_data["VIF"] = [variance_inflation_factor(X.values, i)
                  for i in range(len(X.columns))]
print(vif_data.sort_values(by=['VIF'], ascending=False))
```

	Feature	VIF
6	last_borrowed_in_months	1,222.99
12	n_accounts	3.49
13	n_issues	3.45
8	income	1.18
7	credit_limit	1.14
0	score_3	1.09
5	last_amount_borrowed	1.08
4	risk_rate	1.01
9	ok_since	1.00
3	score_6	1.00
17	reported_income	1.00
1	score_4	1.00
16	external_data_provider_credit_checks_last_year	1.00
2	score_5	1.00
15	external_data_provider_credit_checks_last_month	1.00
10	n_bankruptcies	NaN
11	n_defaulted_loans	NaN
14	external_data_provider_credit_checks_last_2_year	NaN

- Como podemos ver, seguindo as regras do VIF, não há colinearidade alta, há somente moderada, aceitável, portanto, as variáveis independentes não estão correlacionadas.

Exportar Dataset Limpo

```
In [49]: train_no_outliers.to_csv('data/train.csv', index=False)
```

Perguntas de Negócios

- Nessa primeira parte das Perguntas de Negócio optei em criar questões a respeito do Score 3 Médios - 301 até 700, pois analisando a frequência, é onde o Score 3 vai no máximo até o valor de 640.00.

```
In [50]: train_no_outliers.score_3.value_counts().sort_index(ascending=False)
```

```
Out[50]: score_3  
640.00      44  
630.00      67  
620.00      88  
610.00      92  
600.00     104  
590.00     118  
580.00     120  
570.00     164  
560.00     193  
550.00     240  
540.00     284  
530.00     282  
520.00     353  
510.00     400  
500.00     444  
490.00     481  
480.00     542  
470.00     552  
460.00     677  
450.00     781  
440.00     751  
430.00     812  
420.00     883  
410.00     997  
400.00     979  
390.00    1042  
380.00    1063  
370.00    1130  
360.00    1138  
350.00    1231  
340.00    1844  
330.00    1204  
320.00    1193  
310.00    1197  
300.00    1194  
290.00    1103  
280.00    1061  
270.00    1079  
260.00    932  
250.00    931  
240.00    857  
230.00    784  
220.00    680  
210.00    612  
200.00    540  
190.00    477  
180.00    419  
170.00    374  
160.00    276  
150.00    239  
140.00    210  
130.00    139  
120.00    105  
110.00     97  
100.00     59  
 90.00     36  
 80.00     32  
 70.00     14  
 60.00      8
```

```
50.00      2
Name: count, dtype: int64
```

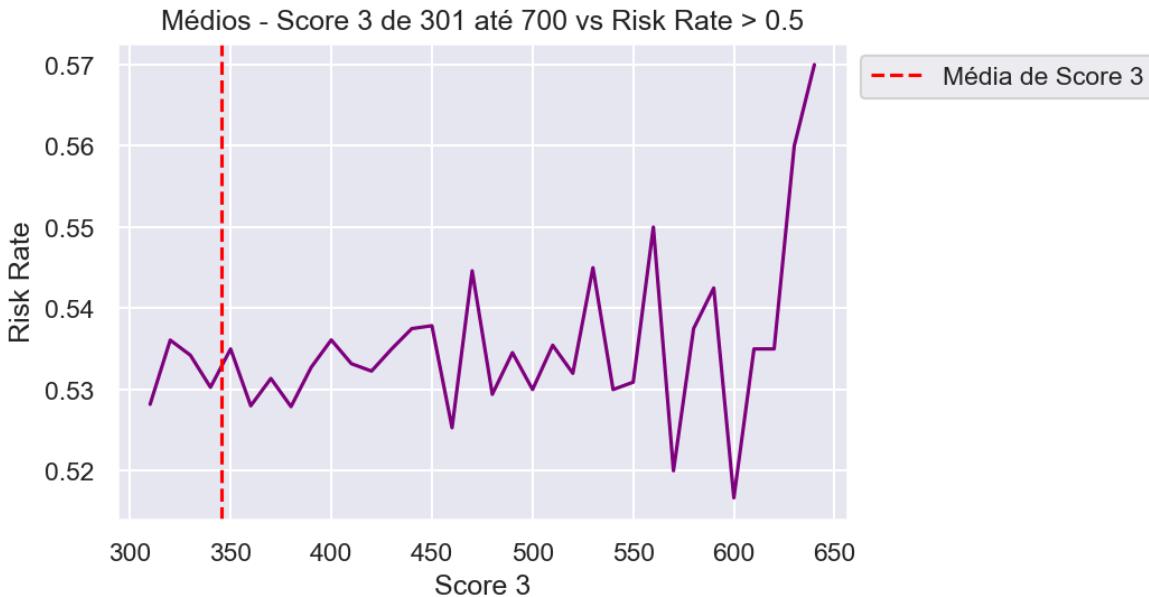
Perguntas sobre o Médios - Score 3 de 301 até 700

- Há clientes "Médios - Score 3 de 301 até 700", que possuem Risk Rate maior do que 0.5?

```
In [51]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.risk_rate > 0.5)])
print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 491 clientes.

```
In [52]: data = train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.last_amount_borrowed > 20000)]
aux1 = data[['score_3', 'risk_rate']].groupby('score_3').mean().reset_index();
score_3_plot(train_no_outliers, "score_3", "risk_rate", "Risk Rate", "Médios - S")
```



R: Sim, foram encontrados 491 clientes com Score 3 entre 301 até 700 e com Risk Rate maior do que 0.5.

- Há clientes "Médios - Score 3 de 301 até 700" que possuem "Último Valor Emprestado"("last_amount_borrowed") maiores que 20.000?

```
In [53]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.last_amount_borrowed > 20000)])
print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 1746 clientes.

```
In [54]: data = train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.last_amount_borrowed > 20000)]
aux1 = data[['score_3', 'last_amount_borrowed']].groupby('score_3').mean().res
```

```
score_3_plot(train_no_outliers, "score_3", "last_amount_borrowed", "Last Amount
```



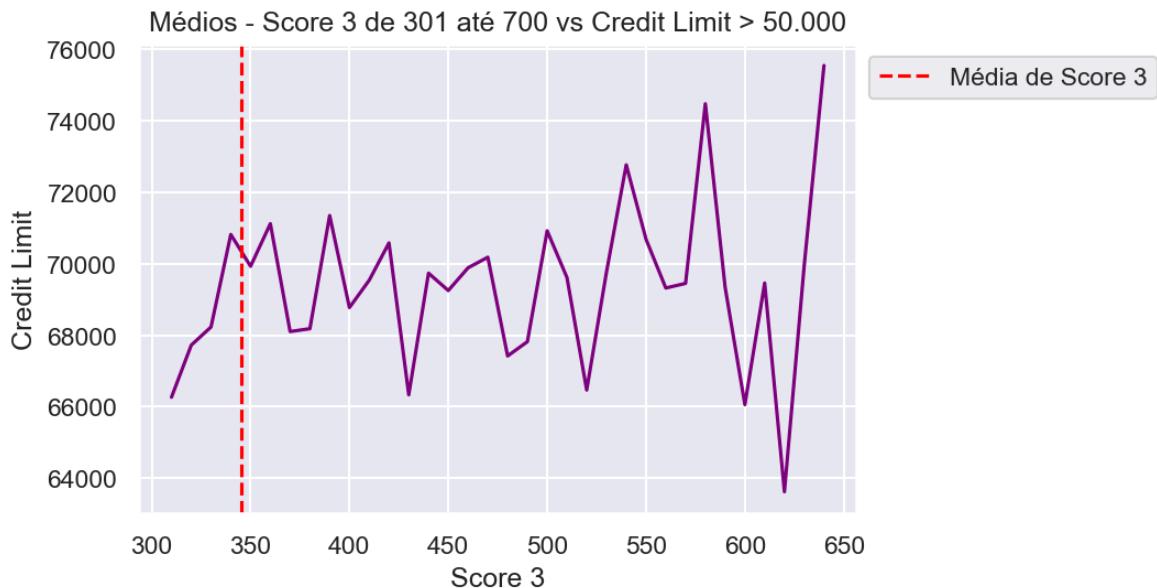
R: Sim, foram encontrados 1746 clientes com Score 3 entre 301 até 700 e com "Último Valor Emprestado"("last_amount_borrowed") maiores que 20.000.

3. Há clientes "Médios - Score 3 de 301 até 700" que possuem "Limite de Crédito" ("credit_limit") maior que 50.000?

```
In [55]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.credit_limit > 50000)][['score_3', 'credit_limit']])
print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 3467 clientes.

```
In [56]: data = train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.credit_limit > 50000)]
aux1 = data[['score_3', 'credit_limit']].groupby('score_3').mean().reset_index()
score_3_plot(train_no_outliers, "score_3", "credit_limit", "Credit Limit", "Médios")
```



R: Sim, foram encontrados 3467 clientes que possuem Score 3 de 301 até 700 e com Limite de Crédito maior que 50.000.

- Há clientes "Médios - Score 3 de 301 até 700" que possuem "Renda"("income") entre 10.000 a 30000.

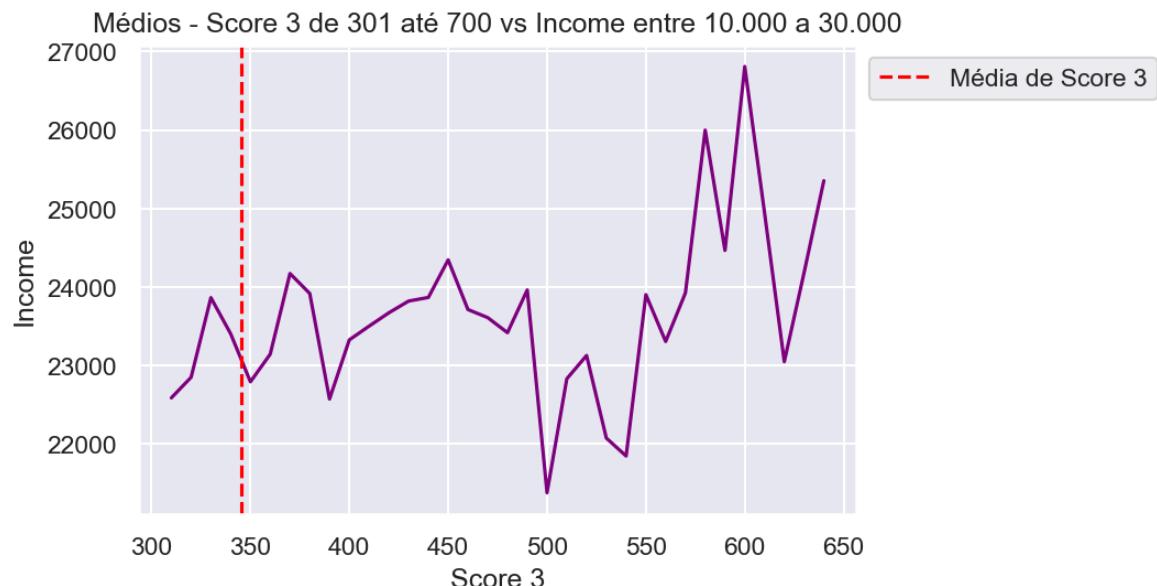
```
In [57]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.income > 10000) & (train_no_outliers.income < 30000)])

print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 1335 clientes.

```
In [58]: data = train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.income > 10000) & (train_no_outliers.income < 30000)]
aux1 = data[['score_3', 'income']].groupby('score_3').mean().reset_index()

score_3_plot(train_no_outliers, "score_3", "income", "Income", "Médios - Score 3")
```



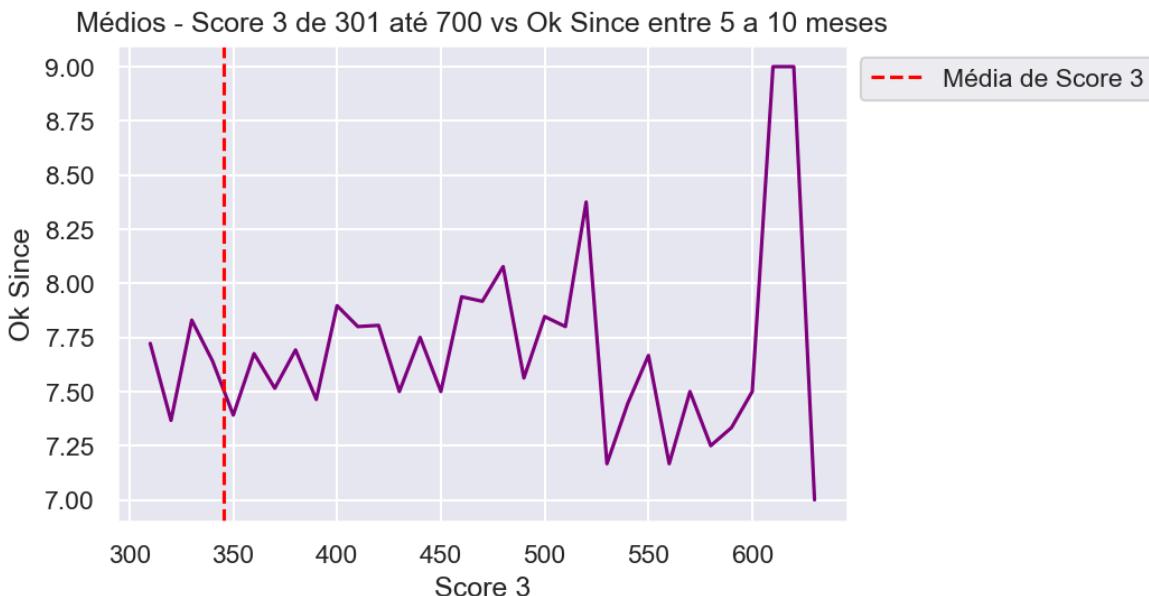
R: Sim, foram encontrados 1335 clientes que possuem Score 3 de 301 até 700 e com Renda entre 10.000 a 30.000.

- Há clientes "Médios - Score 3 de 301 até 700" que possuem "Ok Since" entre 5 meses a 10 meses.

```
In [59]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.ok_since > 5.0) & (train_no_outliers.ok_since < 10.0)])
print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 631 clientes.

```
In [60]: data = train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.ok_since > 5.0) & (train_no_outliers.ok_since < 10.0)]
aux1 = data[['score_3', 'ok_since']].groupby('score_3').mean().reset_index()
score_3_plot(train_no_outliers, "score_3", "ok_since", "Ok Since", "Médios - Score 3 de 301 até 700 vs Ok Since entre 5 a 10 meses")
```



R: Sim, foram encontrados 631 clientes que possuem Score 3 de 301 até 700 e com Ok Since entre 5 meses a 10 meses.

- Há clientes "Médios - Score 3 de 301 até 700" que possuem "Número de Falências" ("n_bankruptcies").

```
In [61]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.n_bankruptcies == 1)])
print("Não, foram encontrados {} clientes.".format(result))
```

Não, foram encontrados 0 clientes.

R: Não, foram encontrados 0 clientes que possuem Score 3 de 301 até 700 e com registros de falência.

7. Há clientes "Médios - Score 3 de 301 até 700" que possuem "Empréstimos que entraram em inadimplência"("n_defaulted_loans").

```
In [62]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.n_defaulted_loans > 0)])
print("Não, foram encontrados {} clientes.".format(result))
```

Não, foram encontrados 0 clientes.

R: Falso. Não há clientes que possuem Score 3 de 301 até 700 com empréstimos que entraram em inadimplência.

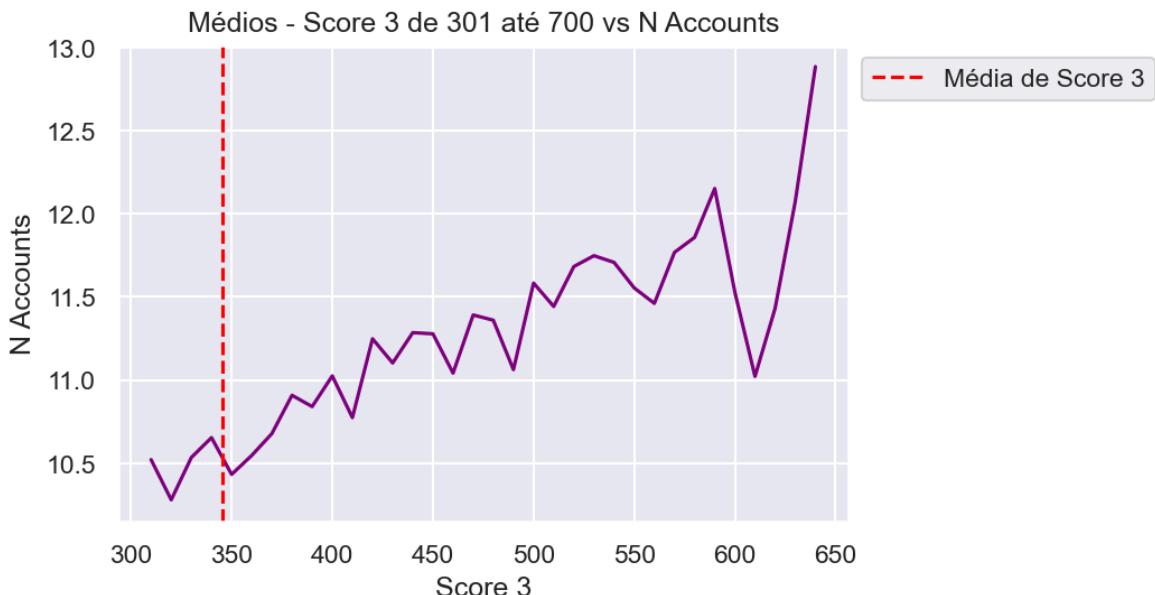
8. Há clientes "Médios - Score 3 de 301 até 700" que possuem "Contas"("n_accounts") maiores que 1.

```
In [63]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.n_accounts > 0)])
print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 21489 clientes.

```
In [64]: data = train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.n_accounts > 0)]
aux1 = data[['score_3', 'n_accounts']].groupby('score_3').mean().reset_index()

score_3_plot(train_no_outliers, "score_3", "n_accounts", "N Accounts", "Médios -")
```



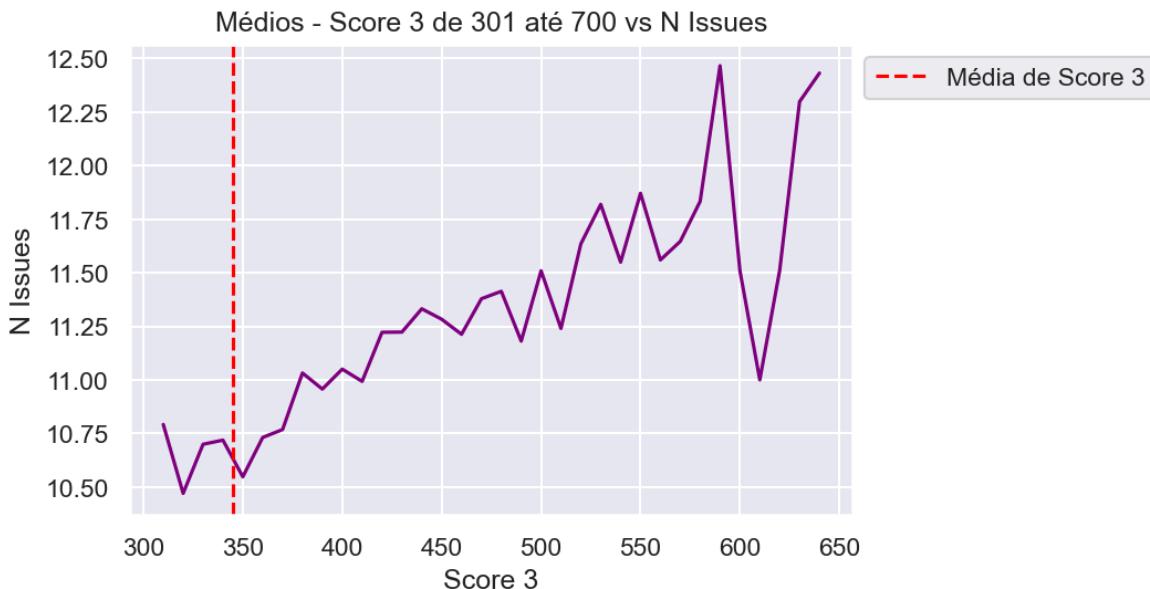
R: Verdadeiro. Há clientes com Score 3 de 301 até 700 que possuem mais de 1 conta.

9. Há clientes "Médios - Score 3 de 301 até 700" que possuem "Número de problemas ou irregularidades"("n_issues") maiores que 1.

```
In [65]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.n_issues > 0)])
print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 21489 clientes.

```
In [66]: data = train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.n_issues > 0)]
aux1 = data[['score_3', 'n_issues']].groupby('score_3').mean().reset_index()
score_3_plot(train_no_outliers, "score_3", "n_issues", "N Issues", "Médios - Sco
```



R: Verdadeiro. Há clientes com Score 3 de 301 até 700 que possuem n_issues maiores do que 1.

10. Há clientes "Médios - Score 3 de 301 até 700" que possuem "Número de vezes que um fornecedor de dados externo realizou verificações de crédito nos últimos 2 anos" ("external_data_provider_credit_checks_last_2_year") maiores que 1?

```
In [67]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.external_data_provider_credit_checks_last_2
print("Não, foram encontrados {} clientes.".format(result))
```

Não, foram encontrados 0 clientes.

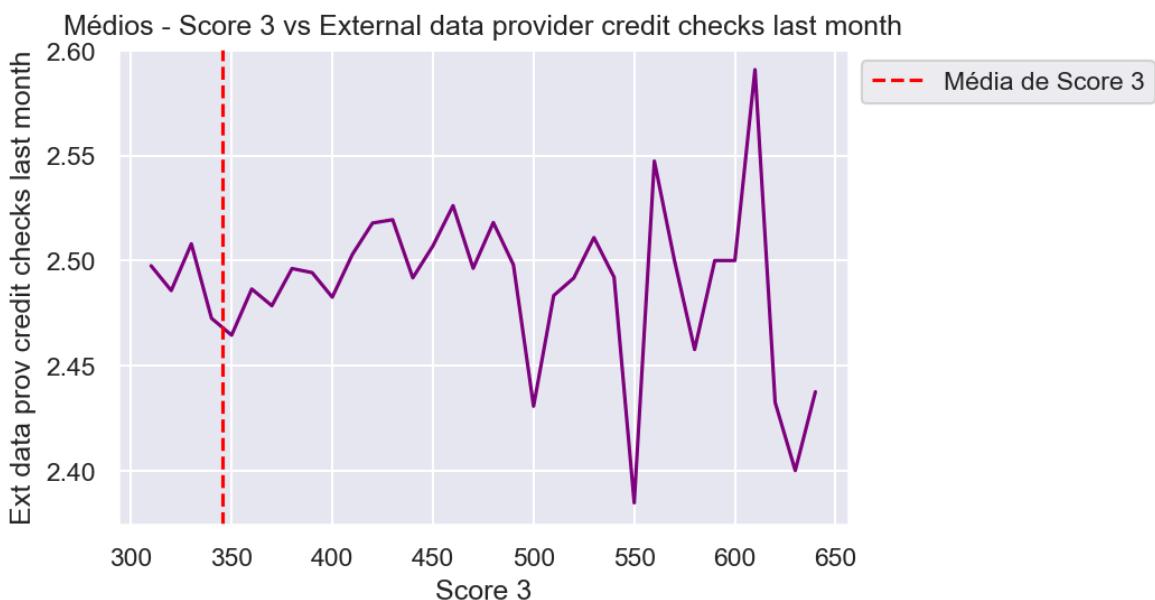
R: Falso. Não há registros.

10. Há clientes "Médios - Score 3 de 301 até 700" que possuem "número de vezes que um provedor de dados externo realizou uma verificação de crédito sobre uma pessoa no último mês."("external_data_provider_credit_checks_last_month") maiores que 1.

```
In [68]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.external_data_provider_credit_checks_last_month > 1)])
print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 10851 clientes.

```
In [69]: data = train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.external_data_provider_credit_checks_last_month > 1)]
aux1 = data[['score_3', 'external_data_provider_credit_checks_last_month']].groupby('score_3').count()
score_3_plot(train_no_outliers, "score_3", "external_data_provider_credit_checks_last_month")
```



R: Verdadeiro. Foram encontrados 10851 clientes com Score 3 de 301 até 700 que possuem external_data_provider_credit_checks_last_month maiores do que 1.

11. Há clientes "Médios - Score 3 de 301 até 700" que possuem "número de vezes que um provedor de dados externo realizou uma verificação de crédito sobre uma pessoa no último ano."("external_data_provider_credit_checks_last_year") igual a 1.

```
In [70]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.external_data_provider_credit_checks_last_year == 1)])
print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 14461 clientes.

```
In [71]: data = train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.external_data_provider_credit_checks_last_year == 1)]
aux1 = data[['score_3', 'external_data_provider_credit_checks_last_year']].groupby('score_3').mean()
score_3_plot(train_no_outliers, "score_3", "external_data_provider_credit_checks_last_year")
```



R: Verdadeiro. Foram encontrados 14461 clientes com Score 3 de 301 até 700 que possuem external_data_provider_credit_checks_last_year igual a 1.

12. Há clientes "Médios - Score 3 de 301 até 700" que possuem "Rendimento Declarado."("reported_income") maior que 100000.

```
In [72]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.score_3 < 700) & (train_no_outliers.reported_income > 100000)])
print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 10993 clientes.

```
In [73]: data = (train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers.score_3 < 700) & (train_no_outliers.reported_income > 100000)]).groupby(['score_3']).mean().reset_index()
aux1 = data[['score_3', 'reported_income']].groupby('score_3').mean().reset_index()
score_3_plot(train_no_outliers, "score_3", "reported_income", "Reported Income", aux1)
```



R: Verdadeiro. Foram encontrados 10993 clientes com Score 3 de 301 até 700 que possuem reported_income maior que 100000.

12. Os clientes "Médios - Score 3 de 301 até 700" estão localizados no estado de SP.

```
In [74]: result = train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outlier
                                (train_no_outliers.shipping_state)]["shipping_state"].value_c
result
```

Out[74]: shipping_state

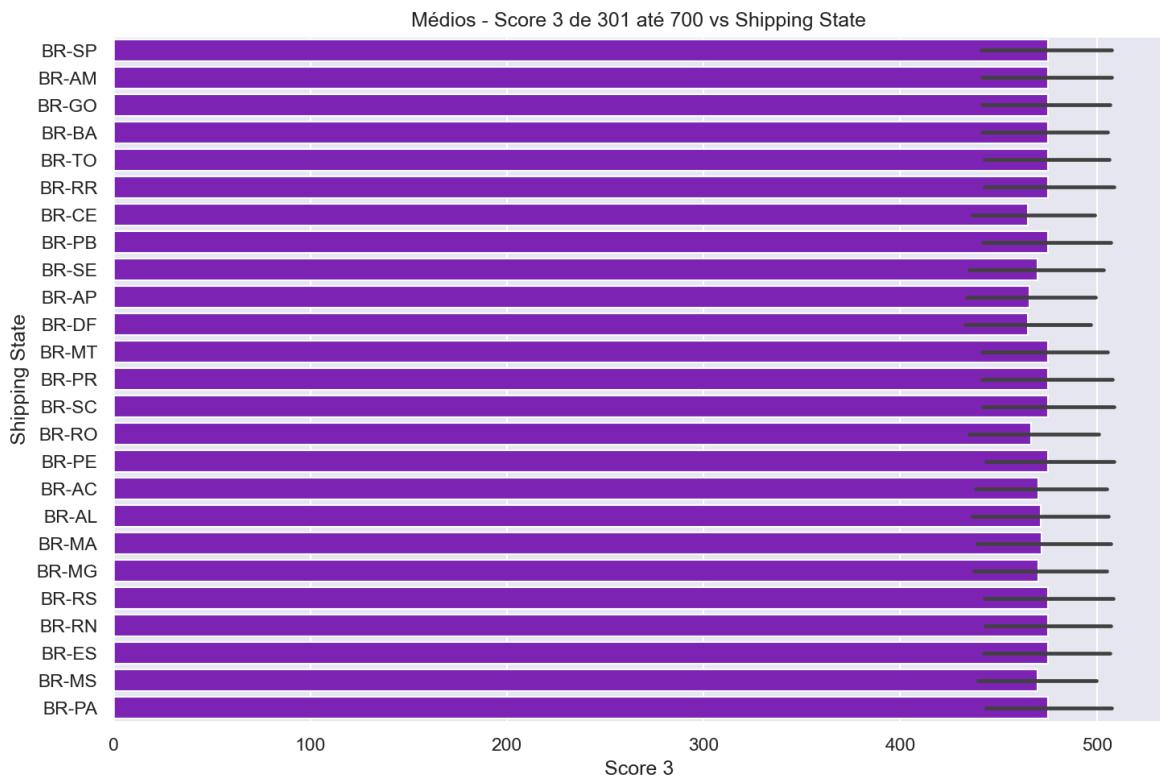
BR-SP	3389
BR-AM	1925
BR-BA	1296
BR-GO	1275
BR-RR	777
BR-MG	733
BR-MA	704
BR-SE	681
BR-PR	672
BR-CE	661
BR-DF	658
BR-AP	655
BR-RO	654
BR-PE	650
BR-RS	644
BR-MS	643
BR-AL	629
BR-TO	628
BR-PB	625
BR-AC	621
BR-PA	610
BR-SC	606
BR-RN	592
BR-ES	583
BR-MT	579

Name: count, dtype: int64

```
In [75]: data = (train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers
                                         (train_no_outliers.shipping_state))]
aux1 = data[['score_3', 'shipping_state']].groupby('score_3').value_counts().r

plt.figure(figsize=(10, 7))
sns.barplot(x='score_3', y='shipping_state', data=aux1, color="#830BD1")

plt.xlabel("Score 3")
plt.ylabel("Shipping State")
plt.title('Médios - Score 3 de 301 até 700 vs Shipping State')
plt.show()
```



Falso. Não só em SP estão os clientes com Score 3 de 301 até 700. Estão em outros estados também.

13. Os clientes "Médios - Score 3 de 301 até 700" não possuem inadimplência.

```
In [76]: result = len(train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers
                                         (train_no_outliers.target_default == 1))]

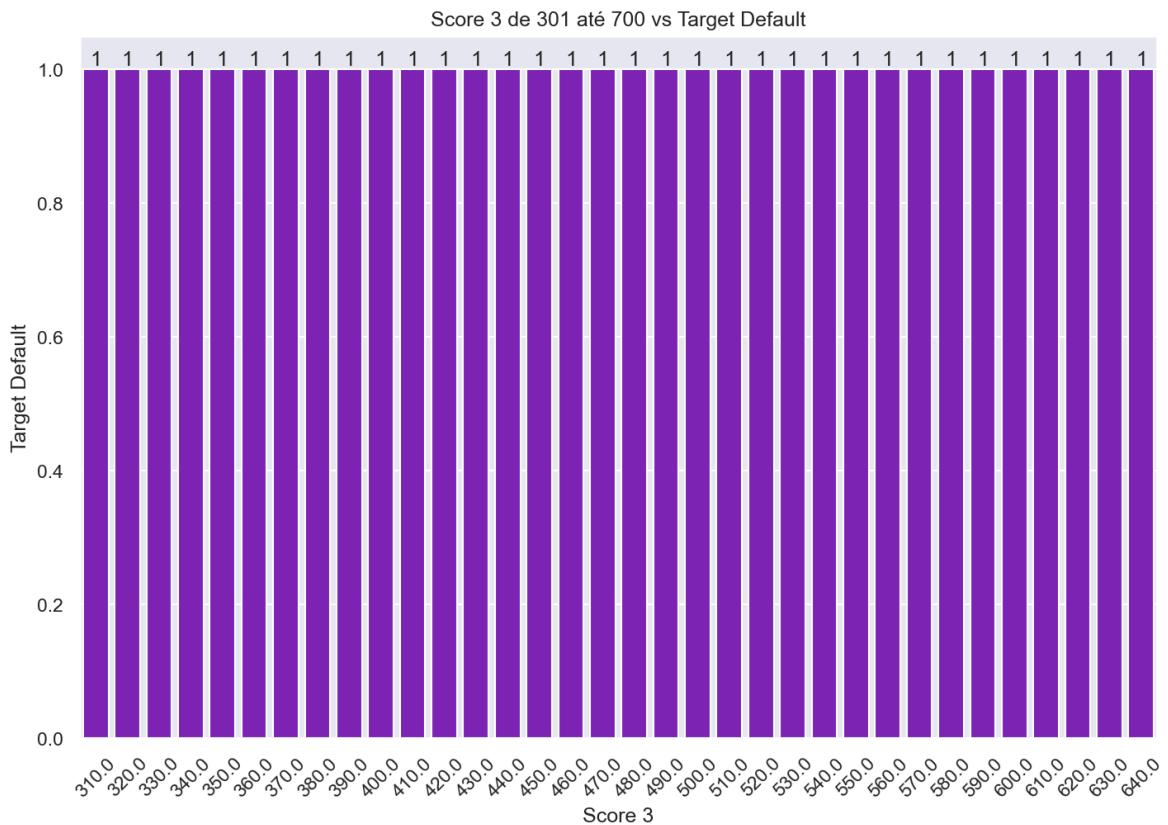
print("Sim, foram encontrados {} clientes.".format(result))
```

Sim, foram encontrados 3392 clientes.

```
In [77]: data = (train_no_outliers[(train_no_outliers.score_3 > 301) & (train_no_outliers
                                         (train_no_outliers.target_default == 1))]
aux1 = data[['score_3', 'target_default']].groupby('score_3').mean().reset_index()

ax = plt.figure(figsize=(10, 7))
ax = sns.barplot(x='score_3', y='target_default', data=aux1, color="#830BD1")
ax.set(xlabel=None, ylabel=None)
for i in ax.containers:
    ax.bar_label(i,)
```

```
ax.tick_params(axis='x', labelrotation=45)
plt.xlabel("Score 3")
plt.ylabel("Target Default")
plt.title('Score 3 de 301 até 700 vs Target Default')
plt.show()
```



R: Falso. Foram encontrados 10 clientes que possuem Score 3 > 700 e que possuem inadimplência (target_default).

Relatório Análise Bivariada

Sobre as correlações feitas com **Diagramas de Dispersão**, **Correlação de Pearson**, e verificação de **Multicolinearidade** e utilização do **VIF(Variance Inflation Factor)**:

- Não foi detectado correlações significativas entre as variáveis utilizando as técnicas mencionadas.
- As Correlações de Pearson ficaram abaixo de **0.50**, apenas duas variáveis tiveram um valor de **0.84**.
- No teste do VIF, os valores de quase todas as variáveis ficaram abaixo de **5.0**, que é o aceitável para a inexistência de multicolinearidade, e correlação, apenas a variável last_borrowed_in_months ficou com valor de 1,222.99, provavelmente é um erro.

Perguntas de Negócio

Foram criadas perguntas de negócio iniciais para os ****Médios - Score 3 de 301 até 700****, que significa um médio risco de inadimplência.

Com isso detectamos as seguintes informações:

- Há clientes com **Médios - Score 3 de 301 até 700** que possuem **Risk Rate maior do que 0.50**, significando um certo risco de inadimplência.
- Há clientes com **Médios - Score 3 de 301 até 700** que fizeram últimos empréstimos maiores que **20.000**.
- Há clientes com **Médios - Score 3 de 301 até 700** com limite de crédito maior que **50.000**.
- Há clientes com **Médios - Score 3 de 301 até 700** com Renda entre **10.000 a 30.000**.
- Há clientes com **Médios - Score 3 de 301 até 700** com **OK Since** entre **5 meses a 10 meses**. Isso significa que há possível inadimplência.
- Há clientes com **Médios - Score 3 de 301 até 700** com 0 registros de falência.
- Há clientes com **Médios - Score 3 de 301 até 700** que possuem mais de **1 conta**.
- Há clientes com **Médios - Score 3 de 301 até 700** que possuem problemas (n_issues) maiores do que **1**.
- Há clientes com **Médios - Score 3 de 301 até 700** que possuem o "número de vezes que um provedor de dados externo realizou uma verificação de crédito sobre uma pessoa no último mês." (**external_data_provider_credit_checks_last_month**) maiores do que **1**. Isso significa que houve checagem na verificação de crédito.
- Há clientes com **Médios - Score 3 de 301 até 700** que possuem "número de vezes que um provedor de dados externo realizou uma verificação de crédito sobre uma pessoa no último ano." (**external_data_provider_credit_checks_last_year**) igual a **1**. Isso também significa que houve checagem na verificação de crédito.
- Há clientes com **Médios - Score 3 de 301 até 700** que possuem Renda Reportada maior que **100.000**.
- Há clientes com **Médios - Score 3 de 301 até 700** que estão em outros estados além de **SP**.
- Há clientes com **Médios - Score 3 de 301 até 700** que possuem inadimplência (**target_default = 1**).

Feature Engineering

Seguindo o que foi feito com as "Perguntas de Negócio utilizando o Score 3", que é o Score mais robusto para o projeto, vamos criar uma nova variável chamada Scores, onde vamos dividir os valores seguindo com base na descrição do Score anteriormente:

O score brasileiro é geralmente dividido em faixas:

- Baixo (0-300): Alto risco de inadimplência.
- Médio (301-700): Risco moderado.
- Alto (701-1000): Baixo risco de inadimplência.

Então vamos ter uma variável chamada de "score" que será dividida com os valores dessa forma e vamos excluir as outras colunas de score para essa nova análise.

```
In [78]: # copia de train
train_no_outliers_2 = train_no_outliers.copy()

In [79]: def scores(score):
    if score <= 300:
        return 'baixo'
    elif score >= 301 and score <= 700:
        return 'medio'
    else:
        return 'alto'

train_no_outliers_2['score'] = train_no_outliers_2['score_3'].apply(scores)

In [80]: # exclusão das colunas de score
exclude_columns = ["score_3", "score_4", "score_5", "score_6"]

train_no_outliers_2.drop(labels = exclude_columns, axis=1, inplace=True)
```

Exportar Dataset Limpo

```
In [81]: train_no_outliers_2.to_csv('data/train_2.csv', index=False)

In [82]: train_no_outliers_2.head()

Out[82]:   risk_rate  last_amount_borrowed  last_borrowed_in_months  credit_limit  income  ok
0      0.26          12,024.02                  36.00     21,968.00  45,013.96
1      0.33          19,237.32                  36.00     40,972.00  80,022.23
2      0.28          12,024.02                  36.00     29,942.00  19,225.52
3      0.28          12,024.02                  36.00     29,942.00  60,043.78
4      0.27          12,024.02                  36.00     29,942.00  45,032.90
```