Atividades previstas para este Notebook:

1. Carregar a tabela olist_ibge_v13

2. Ajustá-lo para ter um modelo aplicado a si.

   Criar variável target.
   Aplicar Label Encoder nas variáveis categóricas.

3. Aplicar Scaling nas variáveis com valores maiores do que 1.

4. Balancear a base utilizando-se do SMOTE over-Sampling

1 - Importação de bibliotecas necessárias

In [1]:

```
import pandas as pd
import numpy as np
pd.options.display.float_format = '{:,.2f}'.format
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
# pd.options.display.max_columns = 100
```

2 - Importação da tabela 'olist_ibge_v13'

In [2]:

```
olist_ibge_v13 = pd.read_excel('olist_ibge_v13.xlsx', sheet_name = "Sheet1", header = 0, in
```

In [3]:

```
olist_ibge_v13.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 92935 entries, 0 to 92934
Data columns (total 19 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Unnamed: 0             92935 non-null  int64  
 1   order_id               92935 non-null  object 
 2   product_id             92935 non-null  object 
 3   seller_id              92935 non-null  object 
 4   product_category_name  92935 non-null  object 
 5   sigla_state            92935 non-null  object 
 6   seller_sigla_state     92935 non-null  object 
 7   review_score           92935 non-null  int64  
 8   qtde_boleto            92935 non-null  int64  
 9   qtde_credit_card       92935 non-null  int64  
 10  qtde_debit_card        92935 non-null  int64  
 11  qtde_voucher           92935 non-null  int64  
 12  soma_payment           92935 non-null  float64
 13  qtde_installments      92935 non-null  int64  
 14  AR_MUN_2018            92935 non-null  float64
 15  POPULAÇÃO ESTIMADA     92935 non-null  int64  
 16  PIB                    92935 non-null  float64
 17  gini                   92935 non-null  float64
 18  dias                   92935 non-null  float64
dtypes: float64(5), int64(8), object(6)
memory usage: 13.5+ MB
```

2.1 - Deletar a coluna 'Unnamed: 0'

> Ela é um ruído que sempre surge ao importarmos um arquivo para um DataFrame

In [4]:

```
olist_ibge_v14 = olist_ibge_v13.drop(['Unnamed: 0'], axis=1)
```

In [5]:

```
olist_ibge_v14.shape, olist_ibge_v13.shape
```

Out[5]:

```
((92935, 18), (92935, 19))
```

In [6]:

```
olist_ibge_v14.head()
```

Out[6]:

	order_id	product_id	
0	50ba38c4dc467baab1ea2c8c7747934d	418d480693f2f01e9cf4568db0346d28	12b9676b00f60f3b700
1	d99e6849f7676dade195f20c26f0eb4f	1081ae52311daac87fb54ba8ce4670ac	4371b634e0efc0e22b09
2	0a9a43ac5fe59c6c4bee2a8f9b9fcce8	c1aabbb6f4caec9f5bf7cd80519d6cc0	579891617139df7d867
3	3f1294f87d79b57f5d55ba7b80c3d94f	0a9b9a871ffaec6c0198334558a6c6a1	f9244d45189d3a360549
4	6c12feac9a308e1382d9b19cca7f20b2	d47821b10559fffaefcf3e57d2b5ff76	0df3984f9dfb3d49ac63

2.2 - Criar coluna target para 'olist_ibge_v14'

> A partir de 'review_score'.
>
> > Chamada 'humor'

In [7]:

```
olist_ibge_v14.columns
```

Out[7]:

```
Index(['order_id', 'product_id', 'seller_id', 'product_category_name',
       'sigla_state', 'seller_sigla_state', 'review_score', 'qtde_boleto',
       'qtde_credit_card', 'qtde_debit_card', 'qtde_voucher', 'soma_paymen
t',
       'qtde_installments', 'AR_MUN_2018', 'POPULAÇÃO ESTIMADA', 'PIB', 'gin
i',
       'dias'],
      dtype='object')
```

In [8]:

```
# Nova coluna chamada 'humor' será 0 (zero), quando review_score for de 1 a 3, ou 1 (um), q
a = {1:0, 2:0, 3:0 , 4:1, 5:1}
olist_ibge_v14['humor'] = olist_ibge_v14['review_score'].map(a)
```

In [9]:

```
olist_ibge_v14.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 92935 entries, 0 to 92934
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   order_id              92935 non-null  object 
 1   product_id            92935 non-null  object 
 2   seller_id             92935 non-null  object 
 3   product_category_name 92935 non-null  object 
 4   sigla_state           92935 non-null  object 
 5   seller_sigla_state    92935 non-null  object 
 6   review_score          92935 non-null  int64  
 7   qtde_boleto           92935 non-null  int64  
 8   qtde_credit_card      92935 non-null  int64  
 9   qtde_debit_card       92935 non-null  int64  
 10  qtde_voucher          92935 non-null  int64  
 11  soma_payment          92935 non-null  float64
 12  qtde_installments     92935 non-null  int64  
 13  AR_MUN_2018           92935 non-null  float64
 14  POPULAÇÃO ESTIMADA    92935 non-null  int64  
 15  PIB                   92935 non-null  float64
 16  gini                  92935 non-null  float64
 17  dias                  92935 non-null  float64
 18  humor                 92935 non-null  int64  
dtypes: float64(5), int64(8), object(6)
memory usage: 13.5+ MB
```

In [10]:

```
olist_ibge_v14['humor'].unique()
```

Out[10]:

```
array([1, 0], dtype=int64)
```

In [11]:

```
qtde_humores = olist_ibge_v14.groupby('humor')['order_id'].count()
```

In [12]:

```
df_qtde_humores = pd.DataFrame(qtde_humores)
```

In [13]:

```
df_qtde_humores
```

Out[13]:

	order_id
humor	
0	19868
1	73067

In [14]:

```
print(73067+19868)
```

92935

In [15]:

```
print(19868*100/92935)
```

21.378382740625167

In [16]:

```
print(73067*100/92935)
```

78.62161725937483

In [17]:

```
olist_ibge_v14.head()
```

Out[17]:

	order_id	product_id	
0	50ba38c4dc467baab1ea2c8c7747934d	418d480693f2f01e9cf4568db0346d28	12b9676b00f60f3b700
1	d99e6849f7676dade195f20c26f0eb4f	1081ae52311daac87fb54ba8ce4670ac	4371b634e0efc0e22b09
2	0a9a43ac5fe59c6c4bee2a8f9b9fcce8	c1aabbb6f4caec9f5bf7cd80519d6cc0	579891617139df7d867
3	3f1294f87d79b57f5d55ba7b80c3d94f	0a9b9a871ffaec6c0198334558a6c6a1	f9244d45189d3a360549
4	6c12feac9a308e1382d9b19cca7f20b2	d47821b10559fffaefcf3e57d2b5ff76	0df3984f9dfb3d49ac63

In [18]:

```
crosstab = pd.crosstab(olist_ibge_v14['dias'], olist_ibge_v14['humor'], margins = True)
```

In [19]:

```
crosstab.head()
```

Out[19]:

humor	0	1	All
dias			
0.53	0	1	1
0.78	1	0	1
0.86	1	0	1
0.86	1	0	1
0.89	0	1	1

In [20]:

```
olist_ibge_v14.describe()
```

Out[20]:

	review_score	qtde_boleto	qtde_credit_card	qtde_debit_card	qtde_voucher	soma_payme
count	92,935.00	92,935.00	92,935.00	92,935.00	92,935.00	92,935
mean	4.15	0.20	0.77	0.02	0.06	160
std	1.29	0.40	0.43	0.12	0.41	219
min	1.00	0.00	0.00	0.00	0.00	0
25%	4.00	0.00	1.00	0.00	0.00	62
50%	5.00	0.00	1.00	0.00	0.00	105
75%	5.00	0.00	1.00	0.00	0.00	177
max	5.00	1.00	2.00	2.00	25.00	13,664

In [21]:

```
# Deletar 'qtde_boleto', por correlação de -0.9 com 'qtde_credit_card', e 'POPULAÇAO ESTIMA
# Deletar 'order_id' por ser somente chave primária e 'review_score', pois já temos oriunda
olist_ibge_v15 = olist_ibge_v14.drop(['qtde_boleto', 'POPULAÇÃO ESTIMADA', 'order_id', 'rev
```

In [22]:

```
olist_ibge_v15.shape, olist_ibge_v14.shape
```

Out[22]:

```
((92935, 15), (92935, 19))
```

In [23]:

```
olist_ibge_v15.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 92935 entries, 0 to 92934
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   product_id             92935 non-null  object 
 1   seller_id              92935 non-null  object 
 2   product_category_name  92935 non-null  object 
 3   sigla_state            92935 non-null  object 
 4   seller_sigla_state     92935 non-null  object 
 5   qtde_credit_card       92935 non-null  int64  
 6   qtde_debit_card        92935 non-null  int64  
 7   qtde_voucher           92935 non-null  int64  
 8   soma_payment           92935 non-null  float64
 9   qtde_installments      92935 non-null  int64  
 10  AR_MUN_2018            92935 non-null  float64
 11  PIB                    92935 non-null  float64
 12  gini                   92935 non-null  float64
 13  dias                   92935 non-null  float64
 14  humor                  92935 non-null  int64  
dtypes: float64(5), int64(5), object(5)
memory usage: 10.6+ MB
```

In [24]:

```
olist_ibge_v15.head()
```

Out[24]:

	product_id	seller_id	product_category_nam
0	418d480693f2f01e9cf4568db0346d28	12b9676b00f60f3b700e83af21824c0e	cool_stu
1	1081ae52311daac87fb54ba8ce4670ac	4371b634e0efc0e22b09b52907d9d469	esporte_laz
2	c1aabbb6f4caec9f5bf7cd80519d6cc0	579891617139df7d8671d373f0669622	livros_interesse_ger
3	0a9b9a871ffaec6c0198334558a6c6a1	f9244d45189d3a3605499abddeade7d5	eletroportate
4	d47821b10559fffaefcf3e57d2b5ff76	0df3984f9dfb3d49ac6366acbd3bbb85	beleza_sauc

Importação da biblioteca LabelEncoder

In [25]:

```
from sklearn.preprocessing import LabelEncoder
```

Chamando o objeto

In [26]:

```
le = LabelEncoder()
```

Verificando o objeto le

Verificando o objeto le

In [27]:

```
le
```

Out[27]:

```
LabelEncoder()
```

Aplicação do objeto aos dados categóricos de 'olist_ibge_v14'

In [28]:

```
le_product_id = le.fit_transform(olist_ibge_v14['product_id'])
```

In [29]:

```
le_seller_id = le.fit_transform(olist_ibge_v14['seller_id'])
```

In [30]:

```
le_product_category_name = le.fit_transform(olist_ibge_v14['product_category_name'])
```

In [31]:

```
le_sigla_state = le.fit_transform(olist_ibge_v14['sigla_state'])
```

In [32]:

```
le_seller_sigla_state = le.fit_transform(olist_ibge_v14['seller_sigla_state'])
```

Exibição do array le_sigla_state

> le_sigla_state é um array.
> Portanto, será necessário transformá-lo em dataframe.
>
> > Aliás, vamos transformar em array os 05 (cinco) conjuntos gerados acima cujos nomes são 'le_qualquer_coisa'

In [33]:

```
le_sigla_state
```

Out[33]:

```
array([10, 10, 10, ...,  9,  9, 23])
```

In [34]:

```
df_le_product_id = pd.DataFrame(le_product_id,columns = ['le_product_id'])
```

In [35]:

```
df_le_seller_id = pd.DataFrame(le_seller_id,columns = ['le_seller_id'])
```

In [36]:

```
df_le_product_category_name = pd.DataFrame(le_product_category_name,columns = ['le_product_
```

In [37]:

```
df_le_sigla_state = pd.DataFrame(le_sigla_state,columns = ['le_sigla_state'])
```

In [38]:

```
df_le_seller_sigla_state = pd.DataFrame(le_seller_sigla_state,columns = ['le_seller_sigla_s
```

Exibição das linhas iniciais dos DataFrame do LabelEncoder.

In [39]:

```
df_le_product_id.head()
```

Out[39]:

	le_product_id
0	7841
1	1976
2	22780
3	1247
4	25080

In [40]:

```
df_le_seller_id.head()
```

Out[40]:

	le_seller_id
0	207
1	769
2	1011
3	2810
4	160

In [41]:

```
df_le_product_category_name.head()
```

Out[41]:

	le_product_category_name
0	26
1	32
2	48
3	31
4	11

In [42]:

```
df_le_sigla_state.head()
```

Out[42]:

	le_sigla_state
0	10
1	10
2	10
3	8
4	10

In [43]:

```
df_le_seller_sigla_state.head()
```

Out[43]:

	le_seller_sigla_state
0	18
1	21
2	15
3	21
4	7

Junção dos dataframes 'le_qualquer_coisa' com 'olist_ibge_vxx'

In [44]:

```
olist_ibge_v16 = pd.merge(olist_ibge_v15, # tabela da esquerda a ser juntada
                         df_le_product_id,  # tabela da direita a ser juntada
                         left_index=True, #If True, use the index (row labels) from the lef
                         right_index=True) #Same usage as left_index for the right DataFram
```

In [45]:

```
olist_ibge_v17 = pd.merge(olist_ibge_v16, df_le_seller_id, left_index=True, right_index=Tru
```

In [46]:

```
olist_ibge_v18 = pd.merge(olist_ibge_v17, df_le_product_category_name, left_index=True, rig
```

In [47]:

```
olist_ibge_v19 = pd.merge(olist_ibge_v18, df_le_sigla_state, left_index=True, right_index=T
```

In [48]:

```
olist_ibge_v20 = pd.merge(olist_ibge_v19, df_le_seller_sigla_state, left_index=True, right_
```

Deleção dos campos que foram codificados

In [49]:

```
olist_ibge_v21 = olist_ibge_v20.drop(columns=['product_id', 'seller_id', 'product_category_
```

In [50]:

```
olist_ibge_v21.shape, olist_ibge_v20.shape, olist_ibge_v15.shape
```

Out[50]:

```
((92935, 15), (92935, 20), (92935, 15))
```

In [51]:

```
olist_ibge_v21.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 92935 entries, 0 to 92934
Data columns (total 15 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   qtde_credit_card           92935 non-null  int64  
 1   qtde_debit_card            92935 non-null  int64  
 2   qtde_voucher               92935 non-null  int64  
 3   soma_payment               92935 non-null  float64
 4   qtde_installments          92935 non-null  int64  
 5   AR_MUN_2018                92935 non-null  float64
 6   PIB                        92935 non-null  float64
 7   gini                       92935 non-null  float64
 8   dias                       92935 non-null  float64
 9   humor                      92935 non-null  int64  
 10  le_product_id              92935 non-null  int32  
 11  le_seller_id               92935 non-null  int32  
 12  le_product_category_name   92935 non-null  int32  
 13  le_sigla_state             92935 non-null  int32  
 14  le_seller_sigla_state      92935 non-null  int32  
dtypes: float64(5), int32(5), int64(5)
memory usage: 8.9 MB
```

In [52]:

```
olist_ibge_v21.head()
```

Out[52]:

	qtde_credit_card	qtde_debit_card	qtde_voucher	soma_payment	qtde_installments	AR_MUN_
0	1	0	0	219.63	10	ε
1	1	0	0	135.59	1	ε
2	0	0	0	58.28	1	ε
3	1	0	0	1,025.52	8	1,C
4	1	0	0	220.97	4	1,ε

In [53]:

```
olist_ibge_v21.isna().sum()
```

Out[53]:

```
qtde_credit_card          0
qtde_debit_card           0
qtde_voucher              0
soma_payment              0
qtde_installments         0
AR_MUN_2018               0
PIB                       0
gini                      0
dias                      0
humor                     0
le_product_id             0
le_seller_id              0
le_product_category_name  0
le_sigla_state            0
le_seller_sigla_state     0
dtype: int64
```

In [54]:

```
print(73067+19868)
```

```
92935
```

In [55]:

```
print(19868*100/92935)
```

```
21.378382740625167
```

In [56]:

```
print(73067*100/92935)
```

```
78.62161725937483
```

SCALING

In [57]:

```
merged = olist_ibge_v21.copy()
```

In [58]:

```
from sklearn.preprocessing import MinMaxScaler

# Scale only columns that have values greater than 1
to_scale = [col for col in olist_ibge_v21.columns if olist_ibge_v21[col].max() > 1]
mms = MinMaxScaler()
scaled = mms.fit_transform(merged[to_scale])
scaled = pd.DataFrame(scaled, columns=to_scale)

# Replace original columns with scaled ones
for col in scaled:
    merged[col] = scaled[col]

merged.head()
```

Out[58]:

	qtde_credit_card	qtde_debit_card	qtde_voucher	soma_payment	qtde_installments	AR_MUN
0	0.50	0.00	0.00	0.02	0.42	
1	0.50	0.00	0.00	0.01	0.04	
2	0.00	0.00	0.00	0.00	0.04	
3	0.50	0.00	0.00	0.08	0.33	
4	0.50	0.00	0.00	0.02	0.17	

In [59]:

```
merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 92935 entries, 0 to 92934
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   qtde_credit_card        92935 non-null  float64
 1   qtde_debit_card         92935 non-null  float64
 2   qtde_voucher            92935 non-null  float64
 3   soma_payment            92935 non-null  float64
 4   qtde_installments       92935 non-null  float64
 5   AR_MUN_2018             92935 non-null  float64
 6   PIB                     92935 non-null  float64
 7   gini                    92935 non-null  float64
 8   dias                    92935 non-null  float64
 9   humor                   92935 non-null  int64  
 10  le_product_id           92935 non-null  float64
 11  le_seller_id            92935 non-null  float64
 12  le_product_category_name  92935 non-null  float64
 13  le_sigla_state          92935 non-null  float64
 14  le_seller_sigla_state   92935 non-null  float64
dtypes: float64(14), int64(1)
memory usage: 10.6 MB
```

In [60]:

```
merged.head()
```

Out[60]:

	qtde_credit_card	qtde_debit_card	qtde_voucher	soma_payment	qtde_installments	AR_MUN
0	0.50	0.00	0.00	0.02	0.42	
1	0.50	0.00	0.00	0.01	0.04	
2	0.00	0.00	0.00	0.00	0.04	
3	0.50	0.00	0.00	0.08	0.33	
4	0.50	0.00	0.00	0.02	0.17	

In [61]:

```
# Parada para avaliação
# O dataframe normalizado é o 'merged', que tem os campos que já tinha valores entre 0 e 1.
# O dataframe 'scaled' tem somente as colunas que foram normalizados e, portanto, tem um nú
```

Separar as explicativas da variável 'target' (variável alvo, a ser prevista).

In [62]:

```
explicativas = merged.drop(columns=['humor'])
target = merged['humor']
```

In [63]:

```
explicativas.head()
```

Out[63]:

	qtde_credit_card	qtde_debit_card	qtde_voucher	soma_payment	qtde_installments	AR_MUN
0	0.50	0.00	0.00	0.02	0.42	
1	0.50	0.00	0.00	0.01	0.04	
2	0.00	0.00	0.00	0.00	0.04	
3	0.50	0.00	0.00	0.08	0.33	
4	0.50	0.00	0.00	0.02	0.17	

In [64]:

```
explicativas.dtypes
```

Out[64]:

```
qtde_credit_card            float64
qtde_debit_card             float64
qtde_voucher                float64
soma_payment                float64
qtde_installments           float64
AR_MUN_2018                 float64
PIB                         float64
gini                        float64
dias                        float64
le_product_id               float64
le_seller_id                float64
le_product_category_name    float64
le_sigla_state              float64
le_seller_sigla_state       float64
dtype: object
```

In [65]:

```
target.head()
```

Out[65]:

```
0    1
1    1
2    1
3    1
4    1
Name: humor, dtype: int64
```

Criação de dataframe com variaveis selecionadas

> Todas as variáveis selecionadas, exceto as de alta cardinalidade.

In [66]:

```
# Rodaremos os modelos com variáveis explicativas definidas no título do notebook.

expl = explicativas[[ 'dias',  'soma_payment',  'le_product_id', 'le_seller_id', 'le_produc

expl.head()
```

Out[66]:

	dias	soma_payment	le_product_id	le_seller_id	le_product_category_name	PIB	AR_MUN_2
0	0.10	0.02	0.26	0.07	0.36	0.00	
1	0.03	0.01	0.07	0.27	0.44	0.00	
2	0.04	0.00	0.76	0.35	0.67	0.00	
3	0.14	0.08	0.04	0.97	0.43	0.00	
4	0.02	0.02	0.83	0.06	0.15	0.00	

In [67]:

```
expl.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 92935 entries, 0 to 92934
Data columns (total 14 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   dias                      92935 non-null  float64
 1   soma_payment              92935 non-null  float64
 2   le_product_id             92935 non-null  float64
 3   le_seller_id              92935 non-null  float64
 4   le_product_category_name  92935 non-null  float64
 5   PIB                       92935 non-null  float64
 6   AR_MUN_2018               92935 non-null  float64
 7   gini                      92935 non-null  float64
 8   qtde_credit_card          92935 non-null  float64
 9   qtde_debit_card           92935 non-null  float64
 10  qtde_voucher              92935 non-null  float64
 11  qtde_installments         92935 non-null  float64
 12  le_sigla_state            92935 non-null  float64
 13  le_seller_sigla_state     92935 non-null  float64
dtypes: float64(14)
memory usage: 9.9 MB
```

Separação em treino e teste

In [68]:

```
from sklearn.model_selection import train_test_split
x_treino, x_teste, y_treino, y_teste = train_test_split(expl,
                                                        target,
                                                        test_size=0.3,
                                                        random_state=196)
```

USO DO SMOTE

SMOTE

In [69]:

```
pip install imbalanced-learn
```

```
Requirement already satisfied: imbalanced-learn in c:\users\gastao\anaconda3
\lib\site-packages (0.8.0)
Requirement already satisfied: scikit-learn>=0.24 in c:\users\gastao\anacond
a3\lib\site-packages (from imbalanced-learn) (0.24.2)
Requirement already satisfied: joblib>=0.11 in c:\users\gastao\anaconda3\lib
\site-packages (from imbalanced-learn) (0.17.0)
Requirement already satisfied: scipy>=0.19.1 in c:\users\gastao\anaconda3\li
b\site-packages (from imbalanced-learn) (1.5.2)
Requirement already satisfied: numpy>=1.13.3 in c:\users\gastao\anaconda3\li
b\site-packages (from imbalanced-learn) (1.19.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\gastao\anaco
nda3\lib\site-packages (from scikit-learn>=0.24->imbalanced-learn) (2.1.0)
Note: you may need to restart the kernel to use updated packages.
```

In [70]:

```
from imblearn.over_sampling import SMOTE

sm = SMOTE(random_state=42)

expl_sm, target_sm = sm.fit_resample(expl, target)

print(f'''Shape de expl antes do SMOTE: {expl.shape}
Shape de expl após SMOTE: {expl_sm.shape}''')

print('\nBalance of positive and negative classes (%):')
target_sm.value_counts(normalize=True) * 100
```

```
Shape de expl antes do SMOTE: (92935, 14)
Shape de expl após SMOTE: (146134, 14)

Balance of positive and negative classes (%):
```

Out[70]:

```
1    50.00
0    50.00
Name: humor, dtype: float64
```

Treinamento e Teste da Base com Smote

Separação em treino e teste

In [71]:

```
from sklearn.model_selection import train_test_split
x_treino, x_teste, y_treino, y_teste = train_test_split(expl_sm,
                                                        target_sm,
                                                        test_size=0.3,
                                                        random_state=196)
```

Gradient Boosting - simplificado

In [72]:

```
#1
gb_dict = {"criterion": ["friedman_mse",  "mae"],
    'random_state': [1967]
}
```

In [73]:

```
gb_dict
```

Out[73]:

```
{'criterion': ['friedman_mse', 'mae'], 'random_state': [1967]}
```

In [74]:

```
#2
from sklearn.model_selection import GridSearchCV
```

In [75]:

```
#3
from sklearn.ensemble import GradientBoostingClassifier
gb = GradientBoostingClassifier(random_state=42)
```

In [76]:

```
gb
```

Out[76]:

```
GradientBoostingClassifier(random_state=42)
```

In [77]:

```
gb_grid = GridSearchCV(estimator=gb,        # parametro a ser utilizado. No caso, Gradient B
                      param_grid=gb_dict,  # nome do dicionario com parametros
                      scoring='accuracy',  # parametro de validação: acurácia
                      cv=10)               # numero de partições do conjunto de treino a se
```

In [78]:

```
gb_grid
```

Out[78]:

```
GridSearchCV(cv=10, estimator=GradientBoostingClassifier(random_state=42),
             param_grid={'criterion': ['friedman_mse', 'mae'],
                         'random_state': [1967]},
             scoring='accuracy')
```

In [79]:

```
#4
gb_grid.fit(x_treino, y_treino)
```

```
C:\Users\gastao\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:1118: Fu
tureWarning: criterion='mae' was deprecated in version 0.24 and will be remo
ved in version 1.1 (renaming of 0.26). Use criterion='friedman_mse' or 'mse'
instead, as trees should use a least-square criterion in Gradient Boosting.
  warnings.warn("criterion='mae' was deprecated in version 0.24 and "
C:\Users\gastao\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:1118: Fu
tureWarning: criterion='mae' was deprecated in version 0.24 and will be remo
ved in version 1.1 (renaming of 0.26). Use criterion='friedman_mse' or 'mse'
instead, as trees should use a least-square criterion in Gradient Boosting.
  warnings.warn("criterion='mae' was deprecated in version 0.24 and "
C:\Users\gastao\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:1118: Fu
tureWarning: criterion='mae' was deprecated in version 0.24 and will be remo
ved in version 1.1 (renaming of 0.26). Use criterion='friedman_mse' or 'mse'
instead, as trees should use a least-square criterion in Gradient Boosting.
  warnings.warn("criterion='mae' was deprecated in version 0.24 and "
C:\Users\gastao\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:1118: Fu
tureWarning: criterion='mae' was deprecated in version 0.24 and will be remo
ved in version 1.1 (renaming of 0.26). Use criterion='friedman_mse' or 'mse'
instead, as trees should use a least-square criterion in Gradient Boosting.
  warnings.warn("criterion='mae' was deprecated in version 0.24 and "
C:\Users\gastao\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:1118: Fu
tureWarning: criterion='mae' was deprecated in version 0.24 and will be remo
ved in version 1.1 (renaming of 0.26). Use criterion='friedman_mse' or 'mse'
instead, as trees should use a least-square criterion in Gradient Boosting.
  warnings.warn("criterion='mae' was deprecated in version 0.24 and "
C:\Users\gastao\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:1118: Fu
tureWarning: criterion='mae' was deprecated in version 0.24 and will be remo
ved in version 1.1 (renaming of 0.26). Use criterion='friedman_mse' or 'mse'
instead, as trees should use a least-square criterion in Gradient Boosting.
  warnings.warn("criterion='mae' was deprecated in version 0.24 and "
C:\Users\gastao\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:1118: Fu
tureWarning: criterion='mae' was deprecated in version 0.24 and will be remo
ved in version 1.1 (renaming of 0.26). Use criterion='friedman_mse' or 'mse'
instead, as trees should use a least-square criterion in Gradient Boosting.
  warnings.warn("criterion='mae' was deprecated in version 0.24 and "
C:\Users\gastao\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:1118: Fu
tureWarning: criterion='mae' was deprecated in version 0.24 and will be remo
ved in version 1.1 (renaming of 0.26). Use criterion='friedman_mse' or 'mse'
instead, as trees should use a least-square criterion in Gradient Boosting.
  warnings.warn("criterion='mae' was deprecated in version 0.24 and "
C:\Users\gastao\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:1118: Fu
tureWarning: criterion='mae' was deprecated in version 0.24 and will be remo
ved in version 1.1 (renaming of 0.26). Use criterion='friedman_mse' or 'mse'
instead, as trees should use a least-square criterion in Gradient Boosting.
  warnings.warn("criterion='mae' was deprecated in version 0.24 and "
C:\Users\gastao\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:1118: Fu
tureWarning: criterion='mae' was deprecated in version 0.24 and will be remo
ved in version 1.1 (renaming of 0.26). Use criterion='friedman_mse' or 'mse'
instead, as trees should use a least-square criterion in Gradient Boosting.
  warnings.warn("criterion='mae' was deprecated in version 0.24 and "
```

Out[79]:

```
GridSearchCV(cv=10, estimator=GradientBoostingClassifier(random_state=42),
             param_grid={'criterion': ['friedman_mse', 'mae'],
```

```
                        'random_state': [1967]},
             scoring='accuracy')
```

In [80]:

```
#5
gb_grid.best_params_
```

Out[80]:

```
{'criterion': 'friedman_mse', 'random_state': 1967}
```

In [81]:

```
gb_grid.best_score_
```

Out[81]:

```
0.7799850376524222
```

Importação da biblioteca - cálculo da acurácia

In [82]:

```
from sklearn.metrics import accuracy_score
```

Acurácia de treino - Gradient Boosting

In [83]:

```
gb_grid.best_score_
```

Out[83]:

```
0.7799850376524222
```

In [84]:

```
acc_gb_treino = accuracy_score(y_treino,
                              gb_grid.predict( # este é o objeto do grid, já com os melhore
                                  x_treino))
acc_gb_treino
```

Out[84]:

```
0.784980399440822
```

Acurácia de teste - Gradient Boosting

A acurácia de teste está muito próxima à acurácia de treino, o que mostra que o modelo está perfomando bem.

In [85]:

```
acc_gb_teste = accuracy_score(y_teste,
                              gb_grid.predict( # este é o objeto do grid, já com os melhore
                                  x_teste))
acc_gb_teste
```

Out[85]:

```
0.7804566501676513
```

In [86]:

```
from sklearn.metrics import classification_report
```

In [87]:

```
print(classification_report(y_treino,gb_grid.predict(x_treino)))
```

```
              precision    recall  f1-score   support

           0       0.86      0.68      0.76     51018
           1       0.74      0.89      0.81     51275

    accuracy                           0.78    102293
   macro avg       0.80      0.78      0.78    102293
weighted avg       0.80      0.78      0.78    102293
```

In [88]:

```
print(classification_report(y_teste,gb_grid.predict(x_teste)))
print ("A acurácia da previsão é ", accuracy_score(y_teste,gb_grid.predict(x_teste)))
```

```
              precision    recall  f1-score   support

           0       0.85      0.68      0.76     22049
           1       0.73      0.88      0.80     21792

    accuracy                           0.78     43841
   macro avg       0.79      0.78      0.78     43841
weighted avg       0.79      0.78      0.78     43841

A acurácia da previsão é  0.7804566501676513
```

In [89]:

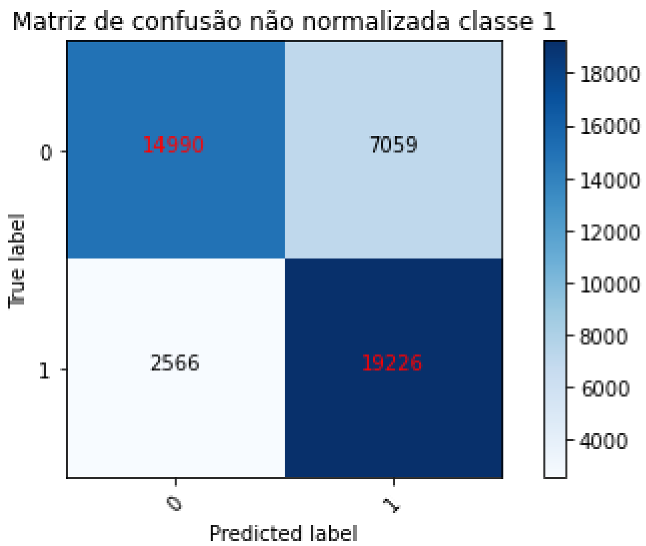
```
#matriz de confusão
from sklearn import metrics
cnf_matrix =  metrics.confusion_matrix(y_teste, gb_grid.predict(x_teste))
```

In [90]:

```
#matriz de confusão
import itertools
from matplotlib import pyplot as plt
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="red" if cm[i, j] > thresh else "black")

    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.tight_layout()
```

In [91]:

```
#visualizando a matriz de confusão
plot_confusion_matrix(cnf_matrix,classes =['0', '1'],
                      title='Matriz de confusão não normalizada classe 1', normalize=False)
```



In [92]:

```
tn = cnf_matrix[0,0]
tp = cnf_matrix[1,1]
fn = cnf_matrix[1,0]
fp = cnf_matrix[0,1]
recall = tp/(tp+fn)
precisão = tp/(tp+fp)
accuracy = (tp+tn)/(tp+tn+fp+fn)
```

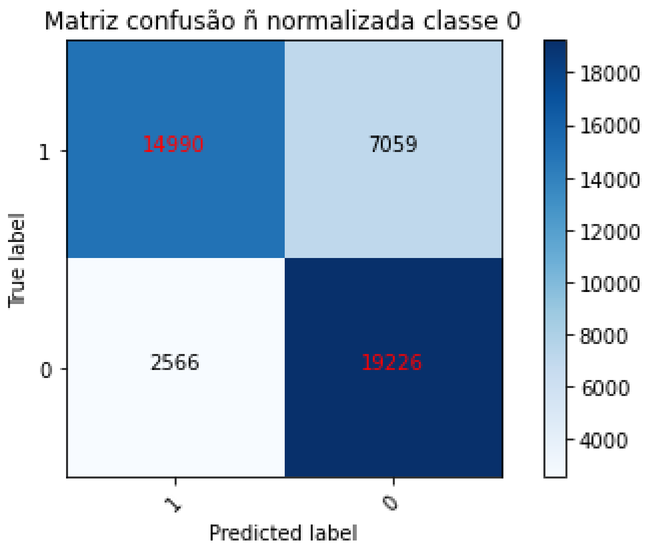
In [93]:

```
print (recall)
print (precisão)
print (accuracy)
```

```
0.8822503671071953
0.731443789233403
0.7804566501676513
```

In [94]:

```
#visualizando a matriz de confusão
plot_confusion_matrix(cnf_matrix,classes =['1', '0'],
                      title='Matriz confusão ñ normalizada classe 0', normalize=False)
```



In [95]:

```
tn = cnf_matrix[1,1]
tp = cnf_matrix[0,0]
fn = cnf_matrix[0,1]
fp = cnf_matrix[1,0]
recall = tp/(tp+fn)
precisão = tp/(tp+fp)
accuracy = (tp+tn)/(tp+tn+fp+fn)
```

Complementação Mário - Matriz de Confusão, Sensibilidade, Especificidade e Feature_importances

In [96]:

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

In [97]:

```
cm1 = confusion_matrix(y_true= y_teste,
                       y_pred= gb_grid.predict(x_teste))
cm1
```

Out[97]:

```
array([[14990,  7059],
       [ 2566, 19226]], dtype=int64)
```

In [98]:

```
sensitivity1 = cm1[0,0]/(cm1[0,0]+cm1[0,1])

specificity1 = cm1[1,1]/(cm1[1,0]+cm1[1,1])
```

In [99]:

```
print('Gradient Boosting com SMOTE')
print('=====================================================')
print('O  recall  da  classe  0  é   : %.2f ' % recall)
print ('A  precisão  da classe 0  é   : %.2f '% precisão)
print ('A  ACURÁCIA DA  PREVISÃO   é  : %.2f '% accuracy)
print('A sensitibilidade da classe 0 : %.2f'% sensitivity1 )
print('A especificidade da classe 0  : %.2f'% specificity1)
```

```
Gradient Boosting com SMOTE
=====================================================
O  recall  da  classe  0  é   : 0.68
A  precisão  da classe 0  é   : 0.85
A  ACURÁCIA DA  PREVISÃO   é  : 0.78
A sensitibilidade da classe 0 : 0.68
A especificidade da classe 0  : 0.88
```

In [100]:

```
gb_grid.best_estimator_.feature_importances_
```

Out[100]:

```
array([0.32959869, 0.00852865, 0.00252828, 0.01009843, 0.1294032 ,
       0.04663085, 0.0098079 , 0.0049921 , 0.00720813, 0.        ,
       0.04390501, 0.28485935, 0.05253168, 0.0699077 ])
```

In [101]:

```
a=pd.concat([pd.Series(x_teste.columns), pd.Series(gb_grid.best_estimator_.feature_importan
```

In [102]:

```
print('Feature Importance')
a
```

```
Feature Importance
```

Out[102]:

	0	1
0	dias	0.33
1	soma_payment	0.01
2	le_product_id	0.00
3	le_seller_id	0.01
4	le_product_category_name	0.13
5	PIB	0.05
6	AR_MUN_2018	0.01
7	gini	0.00
8	qtde_credit_card	0.01
9	qtde_debit_card	0.00
10	qtde_voucher	0.04
11	qtde_installments	0.28
12	le_sigla_state	0.05
13	le_seller_sigla_state	0.07

In [103]:

```
print('Gradient Boosting Feature Importance com SMOTE')
print (a.sort_values(by=1, ascending=False))
```

```
Gradient Boosting Feature Importance com SMOTE
                          0    1
0                      dias 0.33
11        qtde_installments 0.28
4   le_product_category_name 0.13
13     le_seller_sigla_state 0.07
12            le_sigla_state 0.05
5                        PIB 0.05
10              qtde_voucher 0.04
3               le_seller_id 0.01
6                AR_MUN_2018 0.01
1               soma_payment 0.01
8           qtde_credit_card 0.01
7                       gini 0.00
2              le_product_id 0.00
9            qtde_debit_card 0.00
```

In [ ]:

```

```