

Dados

Vamos utilizar como fonte de dados, dois dataset distribuído diretamente para entender o comportamento de compra de vale crédito dos clientes da Bemol.

```
In [4]: import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
import datetime
```

Dados Vale Credito Tabela 1

Primeiramente será observado os dados vale credito tabela 1. Importar os dados para analise.

Os dados tem como colunas: CLIENTES = Id de cada cliente; IDNIVEL = Posição do cliente com relação seu nível; DESCRIAO_NIVEL = Relaciona com a posição do IDNIVEL; SALDO_VC = O Saldo cliente na loja. DT_MUDANCA_PEDRA = ...

```
In [5]: credito1 = pd.read_csv('Dados_vale_credito_tabela1.csv')  
credito1
```

```
Out[5]:
```

	CLIENTE	IDNIVEL	DESCRICAO_NIVEL	SALDO_VC	DT_MUDANCA_PEDRA
0	16923	5	DIAMANTE+	0.68	2020-06-08
1	17873	5	DIAMANTE+	428.98	2020-06-05
2	17974	5	DIAMANTE+	666.77	2020-06-19
3	168775	5	DIAMANTE+	0.00	2020-06-27
4	168962	2	PRATA	0.23	2020-06-02
...
995	219919	5	DIAMANTE+	0.62	2020-06-24
996	1795133	5	DIAMANTE+	2.60	2020-06-14
997	1670911	4	DIAMANTE	26.02	2020-06-16
998	67751	5	DIAMANTE+	0.05	2020-05-21
999	211785	3	OURO	0.16	2020-06-21

1000 rows × 5 columns

Aqui é feito o cálculo da proporção de cada descrição de nível. Nota-se que 44,8% dos clientes são do nível Diamante+, pode ser inferido que aproximadamente metade dos sempre volta para comprar e tornando-se fiel.

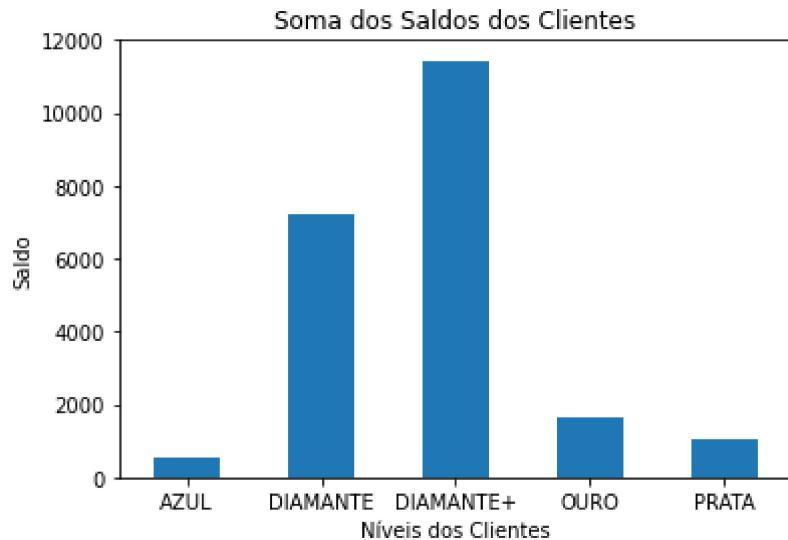
```
In [6]: prop_desc = credito1["DESCRICAO_NIVEL"].value_counts()  
(prop_desc/len(credito1))*100 #Calculo da proporção
```

```
Out[6]: DIAMANTE+    44.8  
DIAMANTE      29.7  
OURO          16.6  
PRATA          7.8  
AZUL           1.1  
Name: DESCRIÇÃO_NÍVEL, dtype: float64
```

O primeiro banco de dados descreve o nível de cliente e seu saldo. Sendo assim, é feito a soma do saldo para cada descrição de nível.

```
In [7]: nivel1 = credito1.groupby(['DESCRICAO_NIVEL'])['SALDO_VC'].sum()  
nivel1  
nivel1.plot.bar()  
  
plt.title('Soma dos Saldos dos Clientes') #Título grafico  
plt.xticks(rotation=0) #rodar o eixo x  
plt.xlabel("Níveis dos Clientes") # Atribuir nome eixo x  
plt.ylabel("Saldo") #Atribuir nome eixo y
```

```
Out[7]: Text(0, 0.5, 'Saldo')
```



Como grande parte dos clientes são Diamante+, conclui-se que o seus saldos serão bem maiores que os outros níveis.

Dados Vale Credito Tabela 2

A segunda análise será com os dados vale credito tabela 2. Importar os dados para analise.

```
In [8]: credito2 = pd.read_csv('Dados_vale_credito_tabela2.csv')
credito2
```

	CLIENTE	DEBITO_OU_CREDITO	TIPO_TRANSACAO_VC	ID_LISTA_PRESENTES	PEDIDO_BOL	VALOR_MOVIMENTADO	SALDO	MOED/
0	3059785	CREDITO	CREDITO_POR_TROCO	0	0	0.37	0.37	BR
1	197026	CREDITO	CREDITO_POR_TROCO	0	0	0.39	3.20	BR
2	2435219	CREDITO	CREDITO_POR_TROCO	0	0	1.15	1.15	BR
3	1736644	CREDITO	CREDITO_POR_TROCO	0	0	0.15	2.66	BR
4	2010537	DEBITO	DEBITO_PAGAMENTO_COM_VC	0	0	-0.50	0.00	BR
...
380319	1046303	CREDITO	CREDITO_POR_TROCO	0	0	0.90	0.90	BR
380320	155516	DEBITO	DEBITO_PAGAMENTO_COM_VC	0	0	-0.20	0.75	BR

CLIENTE	DEBITO_OU_CREDITO		TIPO_TRANSACAO_VC	ID_LISTA_PRESENTES	PEDIDO_BOL	VALOR_MOVIMENTADO	SALDO	MOEDA
380321	2087858	CREDITO	CREDITO_POR_TROCO	0	0	0.23	0.47	BR
380322	2133650	CREDITO	CREDITO_GERADO POR COMPRA DE VC	0	0	85.00	88.35	BR
380323	1452993	CREDITO	CREDITO_POR_TROCO	0	0	0.89	3.95	BR

380324 rows × 12 columns



Observando os resultados abaixo temos que a proporção de pagamento por CREDITO com o tipo de CREDITO_POR_TROCO é 61,75%, é o que mais ocorre durante as compras nas lojas da Bemol. Com a opção de pagamento DEBITO com o tipo de transação DEBITO_PAGAMENTO_COM_VC é um meio menos usado, com proporção de 14,93%.

```
In [9]: pag = credito2.groupby(['DEBITO_OU_CREDITO'])['TIPO_TRANSACAO_VC'].value_counts()
media_pag=pag/len(credito2)*100
media_pag
```

DEBITO_OU_CREDITO	TIPO_TRANSACAO_VC	media_pag
CREDITO	CREDITO_POR_TROCO	61.756555
	CREDITO_GERADO_POR_COMPRA_DE VC	6.325133
	CREDITO_POR_COMPRA_LISTA_PRESENTES	1.788738
	DEBITO_DEVOLUCAO_COMPRA_FEITA_COM VC	0.263985
	CREDITO_PREMIACAO_FUNCIONARIO	0.183265
	CREDITO_PREMIACAO_GERENCIAL_CLIENTES	0.161704
	AJUSTE_CONTABIL_OU_LANCAM_INCORRETO	0.096760
	CREDITO_GERADO_DEVOLUCAO_COMPRA	0.041018
	CREDITO_CONVERSAO_CRED_GERADO_DEVOLUCAO	0.013673
	TRANSFERENCIA_ENTRE_CLIENTES_BEMOL	0.009466
	AJUSTE_LANCAMIENTO_INCORRETO	0.003944
	TRANSFERENCIA_ENTRE_CLIENTES_BEMOL_WHATSAPP	0.003681
	DEBITO_DEVOLUCAO_COMPRA_DE VC	0.001578
	DEBITO_ESTORNO_TROCO	0.000263
	DEBITO_PAGAMENTO_COM VC	14.934635
	DEBITO_COMPRA_COM VC	8.569273
	DEBITO_AUTOMATICO_PAG_PRESTACOES	5.815831
	TRANSFERENCIA_ENTRE_CLIENTES_BEMOL	0.011306
	DEBITO_DEVOLUCAO_COMPRA_DE VC	0.004996
	TRANSFERENCIA_ENTRE_CLIENTES_BEMOL_WHATSAPP	0.004996
DEBITO_ESTORNO_LISTA_PRESENTES	0.004207	
DEBITO_POR_ESTORNO_TROCO	0.004207	
AJUSTE_CONTABIL_OU_LANCAM_INCORRETO	0.000263	
DEBITO_DUPLICIDADE_LANCAMIENTO	0.000263	

DOACAO
Name: TIPO_TRANSACAO_VC, dtype: float64

0.000263

```
In [10]: type(credito2["TIPO_TRANSACAO_VC"])
p = credito2["TIPO_TRANSACAO_VC"].value_counts()
b = (p/len(credito2))*100
b
```

```
Out[10]: CREDITO_POR_TROCO           61.756555
DEBITO_PAGAMENTO_COM_VC            14.934635
DEBITO_COMPRA_COM_VC               8.569273
CREDITO_GERADO_POR_COMPRA_DE_VC   6.325133
DEBITO_AUTOMATICO_PAG_PRESTACOES 5.815831
CREDITO_POR_COMPRA_LISTA_PRESENTES 1.788738
DEBITO_DEVOLUCAO_COMPRA_FEITA_COM_VC 0.263985
CREDITO_PREMIACAO_FUNCIONARIO     0.183265
CREDITO_PREMIACAO_GERENCIAL_CLIENTES 0.161704
AJUSTE_CONTABIL_OU_LANCAM_INCORRETO 0.097023
CREDITO_GERADO_DEVOLUCAO_COMPRA   0.041018
TRANSFERENCIA_ENTRE_CLIENTES_BEMOL 0.020772
CREDITO_CONVERSAO_CRED_GERADO_DEVOLUCAO 0.013673
TRANSFERENCIA_ENTRE_CLIENTES_BEMOL_WHATSAPP 0.008677
DEBITO_DEVOLUCAO_COMPRA_DE_VC     0.006573
DEBITO_POR_ESTORNO_TROCO         0.004207
DEBITO_ESTORNO_LISTA_PRESENTES   0.004207
AJUSTE_LANCAMENTO_INCORRETO      0.003944
DEBITO_ESTORNO_TROCO             0.000263
DEBITO_DUPLICIDADE_LANCAMENTO    0.000263
DOACAO                           0.000263
```

Name: TIPO_TRANSACAO_VC, dtype: float64

```
In [11]: # Separar e criar a coluna DATA_CRIAÇÃO por ano e mes
credito2['DATA_DA_MOVIMENTACAO'] = pd.to_datetime(credito2['DATA_DA_MOVIMENTACAO'],
                                                 format='%Y/%m/%d')
credito2['ANO'] = credito2['DATA_DA_MOVIMENTACAO'].dt.year ## cria coluna por ano
credito2['MES'] = credito2['DATA_DA_MOVIMENTACAO'].dt.month ## cria coluna por mes
```

	CLIENTE	DEBITO_OU_CREDITO	TIPO_TRANSACAO_VC	ID_LISTA_PRESENTES	PEDIDO_BOL	VALOR_MOVIMENTADO	SALDO	MOEDA
0	3059785	CREDITO	CREDITO_POR_TROCO	0	0	0.37	0.37	BR
1	197026	CREDITO	CREDITO_POR_TROCO	0	0	0.39	3.20	BR
2	2435219	CREDITO	CREDITO_POR_TROCO	0	0	1.15	1.15	BR
3	1736644	CREDITO	CREDITO_POR_TROCO	0	0	0.15	2.66	BR

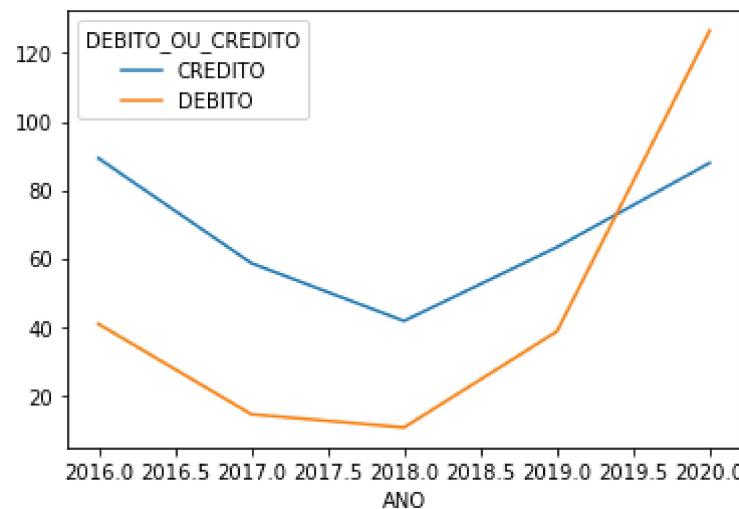
	CLIENTE	DEBITO_OU_CREDITO	TIPO_TRANSACAO_VC	ID_LISTA_PRESENTES	PEDIDO_BOL	VALOR_MOVIMENTADO	SALDO	MOEDA
4	2010537	DEBITO	DEBITO_PAGAMENTO_COM_VC	0	0	-0.50	0.00	BR
...
380319	1046303	CREDITO	CREDITO POR_TROCO	0	0	0.90	0.90	BR
380320	155516	DEBITO	DEBITO_PAGAMENTO_COM_VC	0	0	-0.20	0.75	BR
380321	2087858	CREDITO	CREDITO POR_TROCO	0	0	0.23	0.47	BR
380322	2133650	CREDITO	CREDITO_GERADO_POR_COMPRA_DE_VC	0	0	85.00	88.35	BR
380323	1452993	CREDITO	CREDITO POR_TROCO	0	0	0.89	3.95	BR

380324 rows × 14 columns

Se analisarmos os a coluna DEBITO_OU_CREDITO por ano, é possivel notar que com o passar do ano ocorreu uma queda consideravel no ano de 2018. A partir disso, observa-se que tem uma tendécia de aumento nos anos seguintes.

```
In [13]: credito2.groupby(['ANO','DEBITO_OU_CREDITO'])['SALDO'].mean().unstack().plot()
```

```
Out[13]: <AxesSubplot:xlabel='ANO'>
```



Nessa etapa é coletado os dados dos últimos 6 meses para ver como anda o comportamento.

```
In [14]: seis_m = credito2[(credito2['DATA_CRIACAO'] > '2020-05-19') &
                      (credito2['DATA_CRIACAO'] < '2020-10-18')]
seis_m
```

Out[14]:

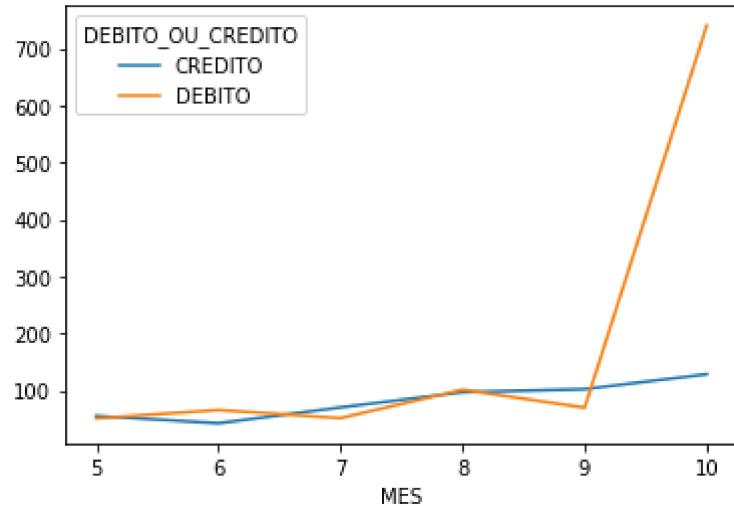
	CLIENTE	DEBITO_OU_CREDITO	TIPO_TRANSACAO_VC	ID_LISTA_PRESENTES	PEDIDO_BOL	VALOR_MOVIMENTADO	SALDO	MOEDA
2	2435219	CREDITO	CREDITO_POR_TROCO	0	0	1.15	1.15	BR
6	1463581	DEBITO	DEBITO_AUTOMATICO_PAG_PRESTACOES	0	0	-2.06	0.00	BR
20	2045570	DEBITO	DEBITO_COMPRA_COM_VC	0	0	-1.02	0.02	BR
30	2584422	DEBITO	DEBITO_AUTOMATICO_PAG_PRESTACOES	0	0	-0.15	0.00	BR
31	1029149	DEBITO	DEBITO_AUTOMATICO_PAG_PRESTACOES	0	0	-0.72	0.00	BR
...
380264	2149641	CREDITO	CREDITO_POR_TROCO	0	0	0.50	1.52	BR
380272	2320696	DEBITO	DEBITO_AUTOMATICO_PAG_PRESTACOES	0	0	-59.00	60.76	BR
380296	1132474	CREDITO	CREDITO_POR_TROCO	0	0	0.59	1.08	BR
380306	2609691	CREDITO	CREDITO_POR_TROCO	0	0	0.13	0.13	BR
380322	2133650	CREDITO	CREDITO_GERADO POR COMPRA DE VC	0	0	85.00	88.35	BR

53845 rows × 14 columns

A análise será feita através das médias do saldo dos clientes por débito e crédito. Como é visto no gráfico abaixo nos últimos 6 meses teve um crescimento de saldo dos clientes, podendo ser possível a compra de mais vale crédito.

```
In [15]: seis_m.groupby(['MES','DEBITO_OU_CREDITO'])['SALDO'].mean().unstack().plot()
plt.ylabel("Saldo") #Atribuir nome eixo y
plt.title('Media de saldo nos últimos 6 meses') #Titulo grafico
```

Out[15]: <AxesSubplot: xlabel='MES'>

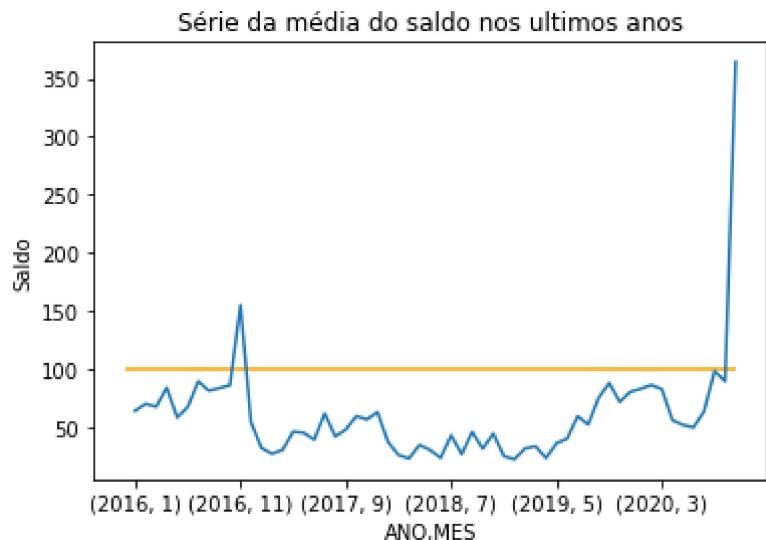


A proxima analise a ser feita é tentar obter uma previsão dos saldo dos clientes. O gráfico abaixo informa uma serie das médias dos ultimos anos por meses, é notório que grande parte dos anos não passava da média 100.

```
In [39]: credito2.groupby(['ANO','MES'])['SALDO'].mean().plot()
plt.hlines(y=100, xmin=-1, xmax=57, color='orange')
#credito2.groupby(['ANO','MES'])['SALDO'].mean() ##resultado das médias

plt.ylabel("Saldo") #Atribuir nome eixo y
plt.title('Série da média do saldo nos ultimos anos') #Titulo grafico
```

```
Out[39]: Text(0.5, 1.0, 'Série da média do saldo nos ultimos anos')
```



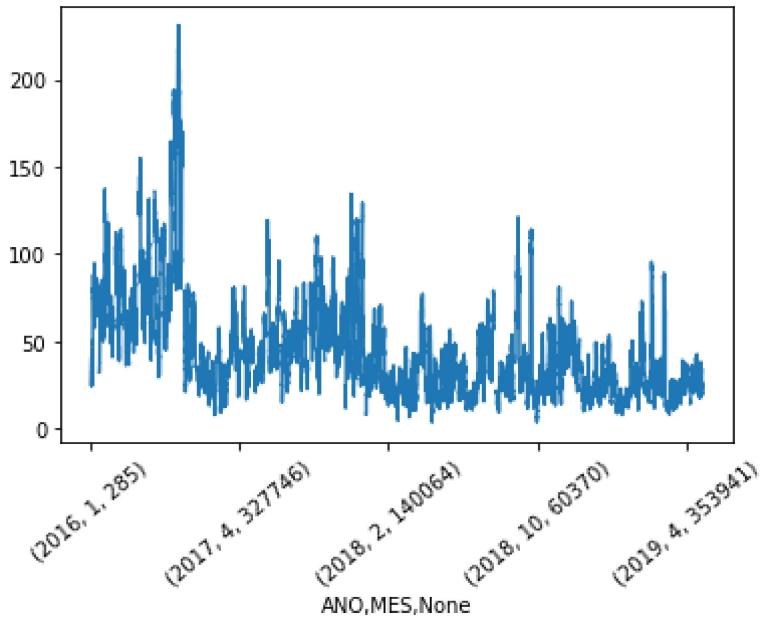
Para realizar a previsão será feita um filtro no ano em um periodo de 02/01/2016-2020/05/18. Deixando uns últimos 6 meses para prever.

```
In [112...]: type(credito2[["ANO", "MES"]])
credito2[["ANO", "MES"]].sum()
```

```
Out[112...]: ANO    767675305
MES     2524760
dtype: int64
```

O método da média móvel foi aplicado ao saldo dos clientes para prever o comportamento nesses períodos.

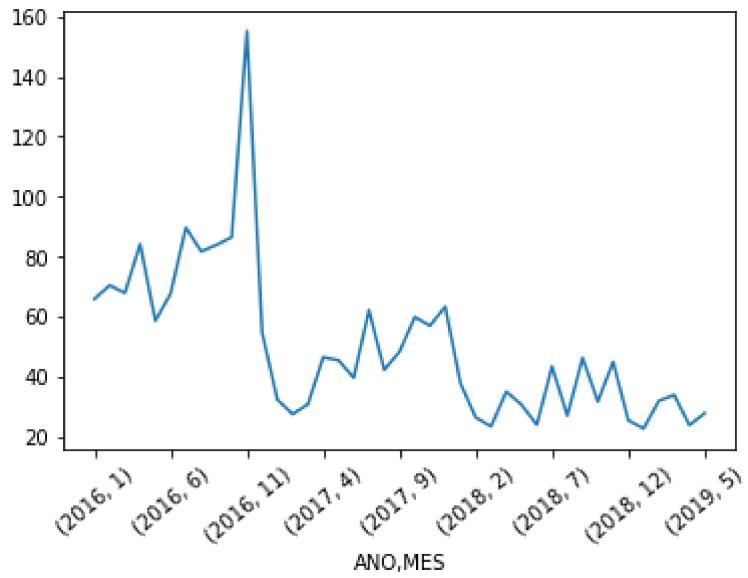
```
In [127...]: pred = credito2[(credito2['DATA_CRIACAO'] > '2016-01-02') &
                     (credito2['DATA_CRIACAO'] < '2019-05-18')]
m = pred.groupby(['ANO', 'MES'])['SALDO'].mean()
mm = pred.groupby(['ANO', 'MES'])['SALDO'].rolling(600).mean().plot(rot=40)
```



O gráfico abaixo mostra as médias do saldo dos clientes no periodo de 2016 a 2019.

In [126...]: `m.plot(rot=40)`

Out[126...]: `<AxesSubplot:xlabel='ANO,MES'>`



In []: Se fizer uma comparação do gráfico das médias móveis e da médias, conclui-se que tem uma estabilização no saldos dos clientes.