Case - Time de Dados e Analytics

Thaís Souza Godoy

1.1. Solução com SQL:

```
-- Verificação da quantidade de linhas da tabela e suas diferenças

SELECT

*

,TB_LOCAL - TB_GCP AS DIFERENCA

FROM( SELECT

(SELECT COUNT(*) FROM application_record_local) AS TB_LOCAL, (SELECT
COUNT(*) FROM application_record_gcp) AS TB_GCP

);
```

Diferença absoluta de linhas entre as tabelas local e GCP:

TB_LOCAL	TB_GCP	DIFERENCA
438.510	437.459	1.051

```
-- Verificação da quantidade de ID distintos da tabela e suas diferenças

SELECT

*

,TB_LOCAL - TB_GCP AS DIFERENCA_ID_DISTINTOS

FROM( SELECT

(SELECT COUNT(DISTINCT ID) FROM application_record_local) AS TB_LOCAL,
(SELECT COUNT(DISTINCT ID) FROM application_record_gcp) AS TB_GCP

);
```

Diferença da quantidade de IDs distintos entre as tabelas:

TB_LOCAL	TB_GCP	DIFERENCA_ID_DISTINTOS
438.510	434.459	4.051

É possível concluir que existe uma diferença absoluta de 1051 linhas de uma tabela para outra. No entanto, há 3000 linhas repetidas na tabela do GCP. Há duplicidade de IDs na tabela GCP, se considerarmos apenas os ID distintos.

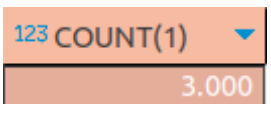
```
-- Lista do ID que tem na tabela local e não tem no GCP

SELECT tb_local.ID

FROM application_record_local tb_local

LEFT JOIN application_record_gcp tb_gcp

ON tb_local.ID = tb_gcp.ID

WHERE tb_gcp.ID IS NULL;
```

Retorna os IDs que tem na tabela local e não na do GCP.

```
-- Lista de ID que tem no GCP e não tem no local

SELECT tb_local.ID

FROM application_record_gcp tb_gcp

LEFT JOIN application_record_local tb_local

ON tb_local.ID = tb_gcp.ID

WHERE tb_local.ID IS NULL;
```

Com o retorno desta consulta, conclui-se que todos os IDs que estão no GCP estão presentes também na tabela local.

```
-- Contagem de IDs duplicados no GCP

SELECT COUNT(1) FROM (

SELECT ID, COUNT(1)

FROM application_record_gcp tb_gcp

GROUP BY ID

HAVING COUNT(1) > 1

)
```



Número de IDs duplicados na tabela do GCP.

```sql
-- IDs duplicados no GCP
SELECT ID, COUNT(1)
FROM application_record_gcp tb_gcp
GROUP BY ID
HAVING COUNT(1) > 1
ORDER BY COUNT(1) DESC
```

Retorna quais IDs estão duplicados e quantas vezes aparecem na tabela.

```sql
-- Análise de padrão das colunas das duas tabelas
SELECT tb_local.*
FROM application_record_local tb_local
LEFT JOIN application_record_gcp tb_gcp
ON tb_local.ID = tb_gcp.ID
WHERE tb_gcp.ID IS NULL;
```

O retorno desta consulta mostra que dos IDs que não estão presentes na tabela GCP, todos possuem o campo FLAG MOBIL como 1. Não há mais informações sobre esse campo para que uma análise mais profunda seja feita.

```sql
SELECT COUNT(1)
FROM application_record_local tb_local
LEFT JOIN (SELECT DISTINCT * FROM application_record_gcp) tb_gcp
ON tb_local.ID = tb_gcp.ID
AND tb_local.CODE_GENDER = tb_gcp.CODE_GENDER

AND tb_local.FLAG_OWN_CAR = tb_gcp.FLAG_OWN_CAR
AND tb_local.FLAG_OWN_REALTY = tb_gcp.FLAG_OWN_REALTY
AND tb_local.CNT_CHILDREN = tb_gcp.CNT_CHILDREN
AND tb_local.AMT_INCOME_TOTAL = tb_gcp.AMT_INCOME_TOTAL
```

```
AND tb_local.NAME_INCOME_TYPE = tb_gcp.NAME_INCOME_TYPE
AND tb_local.NAME_EDUCATION_TYPE = tb_gcp.NAME_EDUCATION_TYPE
AND tb_local.NAME_FAMILY_STATUS = tb_gcp.NAME_FAMILY_STATUS
AND tb_local.NAME_HOUSING_TYPE = tb_gcp.NAME_HOUSING_TYPE
AND tb_local.DAYS_BIRTH = tb_gcp.DAYS_BIRTH
AND tb_local.DAYS_EMPLOYED = tb_gcp.DAYS_EMPLOYED
AND tb_local.FLAG_MOBIL = tb_gcp.FLAG_MOBIL
AND tb_local.FLAG_WORK_PHONE = tb_gcp.FLAG_WORK_PHONE
AND tb_local.FLAG_PHONE = tb_gcp.FLAG_PHONE
AND tb_local.FLAG_EMAIL = tb_gcp.FLAG_EMAIL
AND tb_local.OCCUPATION_TYPE = tb_gcp.OCCUPATION_TYPE
AND tb_local.CNT_FAM_MEMBERS = tb_gcp.CNT_FAM_MEMBERS

WHERE tb_gcp.ID IS NOT NULL;
```

Por meio da consulta ON + AND para cada campo foi possível identificar quais colunas apresentam diferenças entre as tabelas. As colunas CODE GENDER, AMT INCOME TOTAL, DAYS BIRTH e OCCUPATION TYPE tem diferenças de formatação da tabela local para a GCP. A coluna FLAG WORK PHONE apresenta inconsistências na tabela GCP, parte dos dados não foram migrados e não há um padrão que permita algum tipo de tratamento.

Exemplo de inconsistências encontradas nas tabelas:

CODE_GENDER	CODE_GENDER	AMT_INCOME_TOTAL	AMT_INCOME_TOTAL
M	Male	427.500	42.750.000
M	Male	427.500	42.750.000
M	Male	112.500	11.250.000

```
-- Tratando a tabela do GCP
SELECT DISTINCT
ID,
CASE WHEN CODE_GENDER = 'Male' THEN 'M'
WHEN CODE_GENDER = 'Female' THEN 'F'
```

```
ELSE NULL END AS CODE_GENDER,

FLAG_OWN_CAR,

FLAG_OWN_REALTY,

CNT_CHILDREN,

AMT_INCOME_TOTAL/100 AS AMT_INCOME_TOTAL,

NAME_INCOME_TYPE,

NAME_EDUCATION_TYPE,

NAME_FAMILY_STATUS,

NAME_HOUSING_TYPE,

DAYS_BIRTH*(-1) AS DAYS_BIRTH,

DAYS_EMPLOYED,

FLAG_MOBIL,

FLAG_WORK_PHONE,

FLAG_PHONE,

FLAG_EMAIL,

CASE WHEN OCCUPATION_TYPE = 'Without Occupation' THEN ''

ELSE OCCUPATION_TYPE END AS OCCUPATION_TYPE,

CNT_FAM_MEMBERS

FROM application_record_gcp;
```

As diferenças foram tratadas com essa query, para que o dado fosse apresentado da mesma forma que na tabela origem local.

1.2. Solução com Python

**Os desenvolvimentos em Python foram feitos usando Pyspark por conta dos meus estudos atuais na pós-graduação, mas poderiam ser feitos em Pandas.*

O código Python para solucionar o problema foi desenvolvido em um notebook e apresenta os mesmos resultados obtidos pela consulta SQL. Para executar o arquivo `.ipynb`, é necessário ter o PySpark instalado. Além disso, comentários explicativos foram incluídos ao longo do código para facilitar a compreensão.

- verificando a quantidade de linha e diferenças entre as tabelas:

```
DIFERENCA: 1051
TB_LOCAL: 438510
TB_GCP: 437459
```

- verificando a quantidade de IDs distintos da tabela e suas diferenças:

```
TB_GCP_DISTINCT: 434459
TB_LOCAL_DISTINCT: 438510
DIFERENCA_ID_DISTINTOS: 4051
```

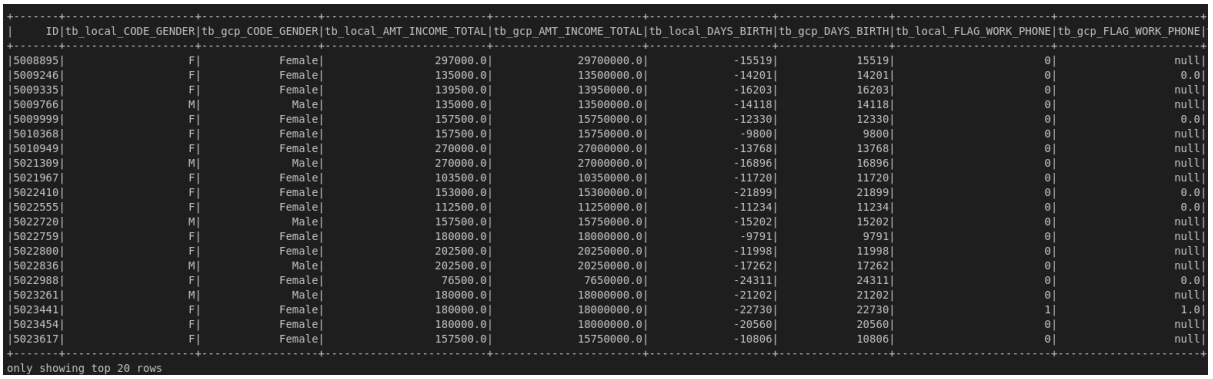
- lista de IDs que aparecem na tabela local e não na GCP:

```
+-------+----------+-----------+--------------+------------+---------------+-----------------+--------------------+-----------------+--------------------+----------+-------------+-----
|     ID|CODE_GENDER|FLAG_OWN_CAR|FLAG_OWN_REALTY|CNT_CHILDREN|AMT_INCOME_TOTAL|  NAME_INCOME_TYPE|NAME_EDUCATION_TYPE|  NAME_FAMILY_STATUS|  NAME_HOUSING_TYPE|DAYS_BIRTH|DAYS_EMPLOYED|FLAG_
+-------+----------+-----------+--------------+------------+---------------+-----------------+--------------------+-----------------+--------------------+----------+-------------+-----
|5033986|         F|          N|             N|           0|       144000.0|        Pensioner|      Lower secondary|            Widow|Municipal apartment|    -21675|       365243|
|5045921|         F|          N|             Y|           0|        40500.0|        Pensioner|      Lower secondary|          Married|  House / apartment|    -23933|       365243|
|5052997|         F|          N|             Y|           0|        94500.0|        Pensioner|      Lower secondary|        Separated|  House / apartment|    -22926|       365243|
|5087743|         F|          N|             Y|           0|       112500.0|        Pensioner|      Lower secondary|          Married|  House / apartment|    -24821|       365243|
|5126743|         F|          N|             N|           0|       135000.0|          Working|      Lower secondary|          Married|   Rented apartment|    -19019|         -43|
|5260593|         F|          N|             N|           0|       225000.0|          Working|      Lower secondary|            Widow|  House / apartment|    -21811|        -2563|
|5354898|         M|          Y|             Y|           1|       247500.0|          Working|      Lower secondary|          Married|  House / apartment|    -20950|         -196|
|5427531|         M|          N|             Y|           0|       135000.0|        Pensioner|      Lower secondary|          Married|  House / apartment|    -24210|       365243|
|5467022|         F|          Y|             N|           0|       112500.0|Commercial associate|      Lower secondary|Single / not married|Municipal apartment|    -12824|         -701|
|5501357|         M|          N|             Y|           0|       135000.0|          Working|      Lower secondary|      Civil marriage|  House / apartment|    -14550|         -516|
|5585006|         M|          N|             Y|           2|       112500.0|Commercial associate|      Lower secondary|          Married|  House / apartment|    -19742|        -4230|
|5702391|         M|          N|             N|           0|       112500.0|          Working|      Lower secondary|          Married|  House / apartment|    -10123|         -190|
|5742776|         F|          Y|             Y|           0|       135000.0|        Pensioner|      Lower secondary|          Married|  House / apartment|    -22486|       365243|
|5872792|         F|          N|             Y|           0|       157500.0|        Pensioner|      Lower secondary|          Married|Municipal apartment|    -22941|       365243|
|5970743|         F|          N|             N|           0|       126000.0|        Pensioner|      Lower secondary|            Widow|  House / apartment|    -21595|       365243|
|6009146|         F|          N|             Y|           0|        60750.0|        Pensioner|      Lower secondary|            Widow|  House / apartment|    -22653|       365243|
|6060382|         M|          N|             N|           4|       270000.0|          Working|      Lower secondary|          Married|  House / apartment|    -14200|         -467|
|6077889|         M|          Y|             Y|           1|       202500.0|          Working|      Lower secondary|          Married|  House / apartment|    -14489|        -2377|
|6088956|         F|          N|             Y|           0|       202500.0|        Pensioner|      Lower secondary|Single / not married|  House / apartment|    -23232|       365243|
|6103539|         F|          N|             Y|           1|       112500.0|   State servant|      Lower secondary|          Married|  House / apartment|    -19238|        -3509|
+-------+----------+-----------+--------------+------------+---------------+-----------------+--------------------+-----------------+--------------------+----------+-------------+-----
only showing top 20 rows
```

- Contagem de IDs duplicados no GCP e ordenação:

```
Número de IDs com mais de uma ocorrência: 3000
+-------+--------+
| gcp_ID|ID_Count|
+-------+--------+
|5008957|       2|
|5009141|       2|
|5009198|       2|
|5009628|       2|
|5010058|       2|
|5010338|       2|
|5010568|       2|
|5010623|       2|
|5010674|       2|
|5010781|       2|
|5010801|       2|
|5010820|       2|
|5018477|       2|
|5021662|       2|
|5021682|       2|
|5021738|       2|
|5021818|       2|
|5021849|       2|
|5021878|       2|
|5021947|       2|
+-------+--------+
only showing top 20 rows
```

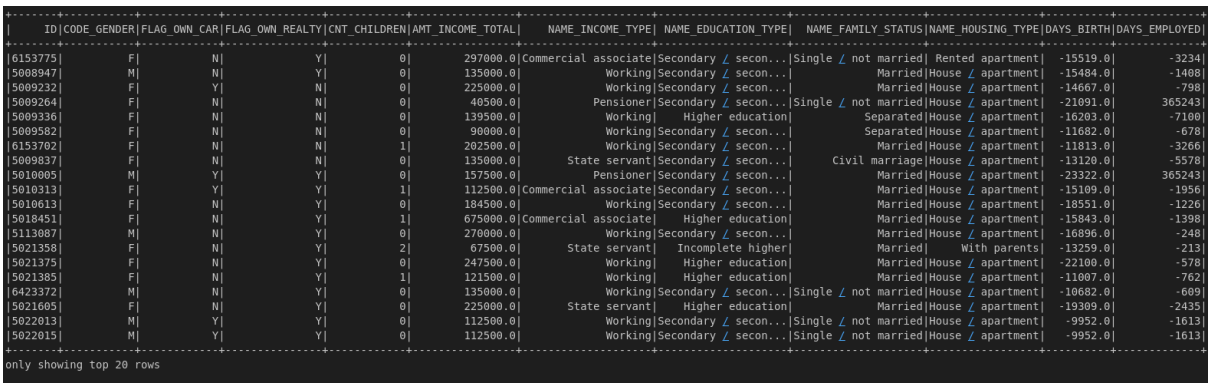
- filtro para identificar as colunas diferentes:



ID	tb_local_CODE_GENDER	tb_gcp_CODE_GENDER	tb_local_AMT_INCOME_TOTAL	tb_gcp_AMT_INCOME_TOTAL	tb_local_DAYS_BIRTH	tb_gcp_DAYS_BIRTH	tb_local_FLAG_WORK_PHONE	tb_gcp_FLAG_WORK_PHONE
5008895	F	Female	297000.0	29700000.0	-15519	15519	0	null
5009246	F	Female	135000.0	13500000.0	-14201	14201	0	0.0
5009335	F	Female	139500.0	13950000.0	-16203	16203	0	null
5009766	M	Male	135000.0	13500000.0	-14118	14118	0	null
5009999	F	Female	157500.0	15750000.0	-12330	12330	0	0.0
5010368	F	Female	157500.0	15750000.0	-9800	9800	0	null
5010949	F	Female	270000.0	27000000.0	-13768	13768	0	null
5021309	M	Male	270000.0	27000000.0	-16896	16896	0	null
5021967	F	Female	103500.0	10350000.0	-11720	11720	0	null
5022410	F	Female	153000.0	15300000.0	-21899	21899	0	0.0
5022555	F	Female	112500.0	11250000.0	-11234	11234	0	0.0
5022720	M	Male	157500.0	15750000.0	-15202	15202	0	null
5022759	F	Female	180000.0	18000000.0	-9791	9791	0	null
5022800	F	Female	202500.0	20250000.0	-11998	11998	0	null
5022836	M	Male	202500.0	20250000.0	-17262	17262	0	null
5022988	F	Female	76500.0	7650000.0	-24311	24311	0	0.0
5023261	M	Male	180000.0	18000000.0	-21202	21202	0	null
5023441	F	Female	180000.0	18000000.0	-22730	22730	1	1.0
5023454	F	Female	180000.0	18000000.0	-20560	20560	0	null
5023617	F	Female	157500.0	15750000.0	-10806	10806	0	null

only showing top 20 rows

- transformação da tabela GCP:



ID	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	NAME_INCOME_TYPE	NAME_EDUCATION_TYPE	NAME_FAMILY_STATUS	NAME_HOUSING_TYPE	DAYS_BIRTH	DAYS_EMPLOYED
6153775	F	N	Y	0	297000.0	Commercial associate	Secondary / secon...	Single / not married	Rented apartment	-15519.0	-3234
5008947	M	N	Y	0	135000.0	Working	Secondary / secon...	Married	House / apartment	-15484.0	-1408
5009232	F	Y	N	0	225000.0	Working	Secondary / secon...	Married	House / apartment	-14667.0	-798
5009264	F	N	N	0	40500.0	Pensioner	Secondary / secon...	Single / not married	House / apartment	-21091.0	365243
5009336	F	N	N	0	139500.0	Working	Higher education	Separated	House / apartment	-16203.0	-7100
5009582	F	N	N	0	90000.0	Working	Secondary / secon...	Separated	House / apartment	-11682.0	-678
6153702	F	N	N	1	202500.0	Working	Secondary / secon...	Married	House / apartment	-11813.0	-3266
5009837	F	N	N	0	135000.0	State servant	Secondary / secon...	Civil marriage	House / apartment	-13120.0	-5578
5010005	M	Y	Y	0	157500.0	Pensioner	Secondary / secon...	Married	House / apartment	-23322.0	365243
5010313	F	Y	Y	1	112500.0	Commercial associate	Secondary / secon...	Married	House / apartment	-15109.0	-1956
5010613	F	N	Y	0	184500.0	Working	Secondary / secon...	Married	House / apartment	-18551.0	-1226
5018451	F	N	Y	1	675000.0	Commercial associate	Higher education	Married	House / apartment	-15843.0	-1398
5113087	M	N	Y	0	270000.0	Working	Secondary / secon...	Married	House / apartment	-16896.0	-248
5021358	F	N	Y	2	67500.0	State servant	Incomplete higher	Married	With parents	-13259.0	-213
5021375	F	N	Y	0	247500.0	Working	Higher education	Married	House / apartment	-22100.0	-578
5021385	F	N	Y	1	121500.0	Working	Higher education	Married	House / apartment	-11007.0	-762
6423372	M	N	Y	0	135000.0	Working	Secondary / secon...	Single / not married	House / apartment	-10682.0	-609
5021605	F	N	Y	0	225000.0	State servant	Higher education	Married	House / apartment	-19309.0	-2435
5022013	M	Y	Y	0	112500.0	Working	Secondary / secon...	Single / not married	House / apartment	-9952.0	-1613
5022015	M	Y	Y	0	112500.0	Working	Secondary / secon...	Single / not married	House / apartment	-9952.0	-1613

only showing top 20 rows

2. Código SQL

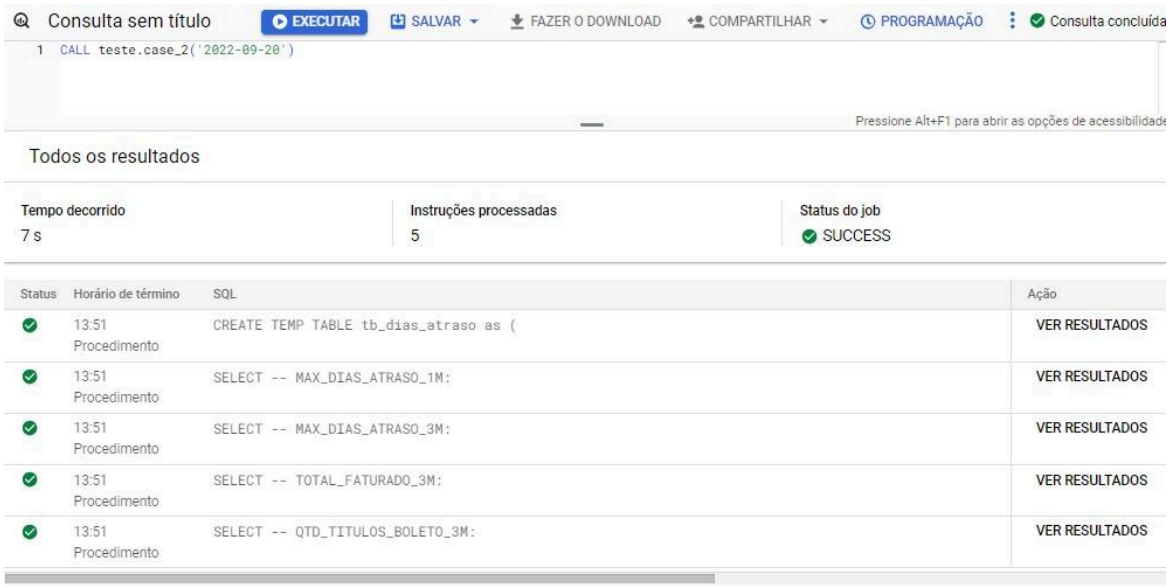
Na primeira questão desenvolvi o código no ambiente Dbeaver + Sqlite, nesta precisei de um ambiente mais robusto para executar a procedure, portanto, desenvolvi a consulta no ambiente do Bigquery.

Para a solução 2 foi necessária a criação de uma procedure, conjunto de instruções que funcionam como uma função encapsulando várias operações. Como a data referência para a análise precisaria ser uma variável, ela foi definida como o parâmetro para a execução da procedure.

Uma tabela temporária foi criada para facilitar a operação das consultas:

```
CREATE TEMP TABLE tb_dias_atraso AS (

SELECT

re.nm_revendedor, IF(dt_pagamento IS
NULL,DATE_DIFF(data_parametro,dt_vencimento,DAY
),MAX(DATE_DIFF(dt_pagamento,dt_vencimento,DAY ))) AS dias_atraso

FROM teste.tb_revendedor re

LEFT JOIN teste.tb_titulos ti

      ON re.id_revendedor = ti.id_revendedor

GROUP BY nm_revendedor,dt_pagamento,dt_vencimento);
```

Como resultado para a data parâmetro 2022-09-20, temos:



- tabela temporária:

Linha	nm_revendedor	dias_atraso
151	Beatriz	30
152	Adriane	31
153	Fernando	31
154	Gabriel	32
155	Mateus	33
156	Mateus	33
157	Gabriel	34
158	Adriane	35
159	Jessica	35
160	Lucas	37
161	Beatriz	40
162	Gabriel	40
163	Mariana	40
164	Lucas	41
165	Jessica	42
166	Lucas	44
167	Lucas	44

- resultado para a consulta máximo dias de atraso em 1 mês:

INFORMAÇÕES DO JOB | RESULTADOS | GRÁFICO | JSON

Linha	nm_revendedor	MAX_DIAS_ATRASO
1	Ariel	30
2	Lucas	23
3	Mateus	26
4	Renata	30
5	Adriane	25
6	Beatriz	30
7	Gabriel	21
8	Jessica	29
9	Mariana	28
10	Fernando	28

- resultado para a consulta máximo dias de atraso para 3 meses:

INFORMAÇÕES DO JOB | RESULTADOS | GRÁFICO

Linha	nm_revendedor	MAX_DIAS_ATRASO
1	Ariel	55
2	Lucas	44
3	Mateus	58
4	Renata	54
5	Adriane	59
6	Beatriz	47
7	Gabriel	57
8	Jessica	53
9	Mariana	58
10	Fernando	54

- total faturado em 3 meses:

INFORMAÇÕES DO JOB | RESULTADOS | GRÁFICO

Linha	nm_revendedor	vlr_total
1	Lucas	6890.6
2	Fernando	5493.4
3	Renata	7959.1
4	Mariana	4797.0
5	Beatriz	9889.7
6	Mateus	4834.1
7	Gabriel	8744.4
8	Ariel	6248.0
9	Adriane	8867.5
10	Jessica	6955.2

- quantidade de boletos a prazo em 3 meses:

Resultados da consulta

INFORMAÇÕES DO JOB | RESULTADOS | GRÁFICO

Linha	nm_revendedor	qtde_titulo
1	Lucas	2
2	Fernando	4
3	Renata	4
4	Mariana	4
5	Beatriz	2
6	Mateus	2
7	Ariel	2
8	Adriane	6
9	Jessica	2

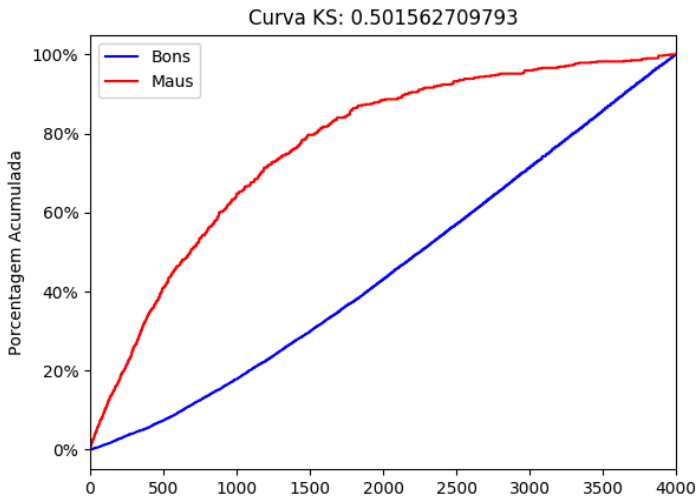
3. Código Python

No problema 3 foi implementada uma função que calcula a métrica KS (Kolmogorov-Smirnov). Medida usada para verificar a eficácia de um modelo de classificação, comparando as distribuições acumuladas de duas classes, a fim de identificar o maior desvio entre elas.

O código foi feito para ler um arquivo, passado como parâmetro para a função. Alguns outros parâmetros como nome das colunas também poderiam ser parametrizados, mas considerei que todos os arquivos possuem a mesma formatação.

É feita a contagem de 'BOM' e 'MAU'. Pelos valores do 'SCORE' foi aberta uma janela de ordenação para o cálculo da porcentagem acumulada da classificação binária de bons e maus.

O valor de KS foi calculado pela diferença absoluta entre os valores acumulados de 'BOM' e 'MAU'.



4. Código Python

No problema 4 foi desenvolvido um script 100% automatizado que retorna a tabela filtrada para o ano de 2023 e 2024:

CO_ANO	CO_MES	CO_NCM	CO_UNID	CO_PAIS	SG_UF_NCM	CO_VIA	CO_URF	QT_ESTAT	KG_LIQUIDO	VL_FOB	VL_FRETE	VL_SEGURO
2023	01	33030010	10	275	SP	1	0817800	333	333	7799.0	469	22
2023	11	33030010	10	275	SP	1	0817800	807	807	23778.0	107	34
2023	09	33030010	10	275	SP	1	0817800	24	24	986.0	4	1
2023	06	33030010	10	275	SP	1	0817800	1024	1024	29167.0	236	42
2023	03	33030010	10	275	SP	1	0817800	544	544	14606.0	172	37
2023	05	33030010	10	275	SP	1	0817800	164	164	4229.0	55	10
2023	04	33030010	10	275	SP	1	0817800	500	500	12619.0	209	31
2023	10	33030010	10	275	SP	1	0817800	76	76	3319.0	11	5
2023	12	33030010	10	275	SP	1	0817800	393	393	9909.0	55	14
2023	08	33030010	10	275	SP	1	0817800	499	499	9114.0	96	13
2023	07	33030010	10	275	SP	1	0817800	626	626	29725.0	136	43

CO_ANO	CO_MES	CO_NCM	CO_UNID	CO_PAIS	SG_UF_NCM	CO_VIA	CO_URF	QT_ESTAT	KG_LIQUIDO	VL_FOB	VL_FRETE	VL_SEGURO
2024	02	33030010	10	275	SP	1	0817800	155	155	5076.0	33	7
2024	07	33030010	10	275	SP	1	0817800	1076	1076	24415.0	120	35
2024	06	33030010	10	275	SP	1	0817800	126	126	4542.0	24	7
2024	04	33030010	10	275	SP	1	0817800	23	23	225.0	5	0
2024	05	33030010	10	275	SP	1	0817800	1014	1014	33592.0	105	48
2024	03	33030010	10	275	SP	1	0817800	58	58	2494.0	8	4
2024	01	33030010	10	275	SP	1	0817800	282	282	6311.0	83	9
2024	08	33030010	10	275	SP	1	0817800	938	938	41757.0	91	60

O script pode ser reexecutado a cada mês já que ainda não finalizamos o ano de 2024. O código faz o download da tabela direto do site do Governo, foi necessário um tratamento para essa primeira parte, porque o site está apresentando um problema de SSL. A solução temporária foi desabilitar a verificação do SSL, não é algo recomendado, mas impossibilitaria o script ser totalmente automático.

Após baixar o arquivo, o script faz o tratamento de algumas colunas, transformando-as em tipo int ou float, filtra o dataframe de acordo com os requisitos: país França, via navio, produto 33030010 e negociação no estado de SP.

As colunas **KG_LIQUIDO** e **VL_FOB** foram somadas, e a média anual do valor da negociação foi calculada. Mantive os valores em dólar para preservar a moeda de referência utilizada nas transações e evitar as flutuações cambiais do real. Isso garante uma análise mais precisa e consistente do valor negociado, evitando distorções que poderiam ocorrer, como no caso de variações no câmbio em 2023, por exemplo.

Média em 2023: U$29,11

Média em 2024: U$32,23