Engenharia de Dados

Cessão de Crédito Consignado

Autor:Alexander Almeida

Última atualização: segunda-feira, 1 de julho de 2024

ÍNDICE

1. MVP

1.1. Objetivo

O objetivo principal é analisar os dados para ser mais assertivo na venda do crédito consignado. A regra principal consiste em ser menor que 85 anos de idade, com margem de saldo superior à 35% dos parcelamentos vigentes.

O dataset possui 6176 instâncias e 9 atributos, com um mix de tipos de dados categóricos e numéricos.

1.2. Análise

- Qual o volume de Idade por UF?
- Qual o volume da Margem de Saldo por UF?
- Qual a relação entre UF, Idade e Margem de Saldo?

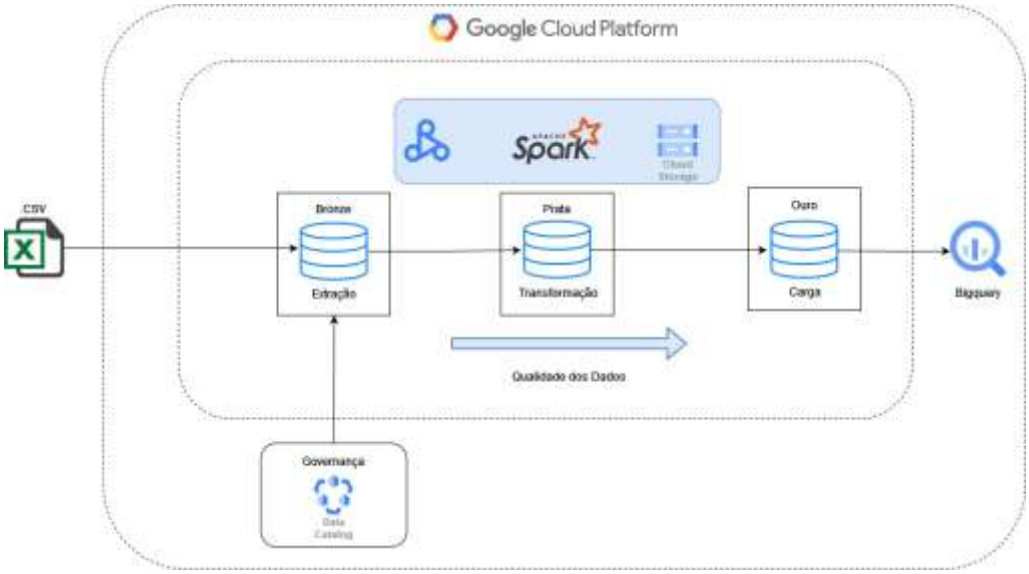
Com estas definições, poderemos montar ações estratégicas para cessão de crédito

1.3. Metadado

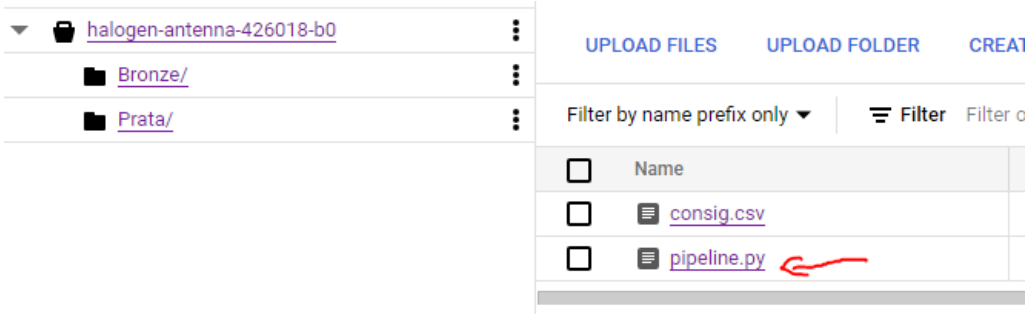
	Field name	Type	Mode	Key	Collation	Default value	Policy tags	Description
☐	cpf	STRING	NULLABLE	-	-	-	-	CPF do beneficiário
☐	nome	STRING	NULLABLE	-	-	-	-	Nome do beneficiário
☐	sexo	STRING	NULLABLE	-	-	-	-	Sexo do beneficiário
☐	idade	INTEGER	NULLABLE	-	-	-	-	Idade do beneficiário
☐	uf	STRING	NULLABLE	-	-	-	-	Unidade Federativa de origem do beneficiário
☐	situacao_funcional	STRING	NULLABLE	-	-	-	-	Situação funcional: Aposentado/Pensionista
☐	tipo_contrato	STRING	NULLABLE	-	-	-	-	Extrato: Aposentadoria Extrato / Não Extrato: Pode perder a aposentadoria
☐	prst_valor	FLOAT	NULLABLE	-	-	-	-	Valor total dos parcelamentos vigentes
☐	margem_saldo	FLOAT	NULLABLE	-	-	-	-	Valor total disponível para cessão de crédito
☐	conceder	INTEGER	NULLABLE	-	-	-	-	Regra do negócio para cessão de crédito, onde 0?CONCEDER e 1?NÃO conceder

1.4. Arquitetura do ETL

Estaremos utilizando a arquitetura medalhão neste MVP, conforme figura abaixo:



Toda construção é realizada através de uma pipeline em pyspark (pipeline.py)



1.4.1. Camada Bronze (Extração)

Na camada bronze carreguei o arquivo .CSV no Cloud Storage GCP, pasta Bronze, em seu estado bruto. A base utilizada é particular e real, mas para apresentação neste MVP, foi necessária a anonimização do campo CPF. Esse processo de extração não foi automatizado, devido a limitação da conta "free" na GCP. Desta forma, realizei o UPLOAD manualmente, conforme figura abaixo:

Referências:
pipeline.py:
https://raw.githubusercontent.com/AlexanderAlmeida/pos_graduacao/master/pipeline.py

Consig.csv:
https://raw.githubusercontent.com/AlexanderAlmeida/pos_graduacao/master/consig.csv

1.4.2. Camada Prata (Transformação)

Na camada prata carreguei o dataframe com spark para início dos tratamentos de dados. Criei um contador de instâncias para observabilidade.

```
In [3]: ############################
        # CAMADA BRONZE (Ingestão)
        ############################

        # Leia o dataset do Cloud Storage
        df_fe = spark.read.csv("gs://halogen-antenna-426018-b0/Bronze/consig.csv",header=True, inferSchema=True)
        df_fe.show()

        # Contando o número de linhas do dataset para observabilidade
        num_rows_csv = df_fe.count()
        print(f"Number of rows in CSV: {num_rows_csv}")
```

```
|0000***2074|CRISTIANO SANTOS ...|MASCULINO|  37| RS|   ATIVO PERMANENTE|     ESTAVEL|             45.07|           1361.11|
0|
|0000***3748|MARIA DA GLORIA V...|FEMININO|  52| RJ|   ATIVO PERMANENTE|     ESTAVEL|             37.15|           2192.17|
0|
|0000***3204|CEREJA KAZUKO NAK...|FEMININO|  80| RS|         APOSENTADO|     ESTAVEL|            1922.6|           5160.92|
0|
|0000***3155|SILBERTO DOS SANT...|MASCULINO|  36| MG|   ATIVO PERMANENTE|     ESTAVEL| 748.4200000000001|           2864.08|
0|
|0000***1234|IRIS PEDRO DE OLI...|FEMININO|  75| GO|         APOSENTADO|     ESTAVEL|            6721.39|          19903.32|
0|
|0000***7742| JANICE CAMPOS MOTTA|FEMININO|  55| RJ|         APOSENTADO|     ESTAVEL| 576.3299999999999|           4161.24|
0|
|0000***4706|FABIO DE JESUS RI...|MASCULINO|  52| RJ|   ATIVO PERMANENTE|     ESTAVEL|            150.29|           3251.0|
0|
|0000***6721|GIOVANI AZEVEDO S...|MASCULINO|  56| SC|     NAO INFORMADO|     ESTAVEL|             119.0|            598.8|
0|
|0000***2700|JERONIMO DOS SANT...|MASCULINO|  58| RJ|   ATIVO PERMANENTE|     ESTAVEL|           1862.14|2880.3500000000004|
0|
|0000***0580|MARCELLO PORTELA ...|MASCULINO|  40| MS|   ATIVO PERMANENTE|     ESTAVEL|            831.53|          11245.95|
0|
|0000***4353|JOSE WEVERGTHON A...|MASCULINO|  82| SP|         APOSENTADO|     ESTAVEL|            683.21|          15200.95|
0|
|0000***0065|VANDERLEI DOMINGU...|MASCULINO|  45| GO|CELETISTA EMPREGADO|     ESTAVEL|1640.6699999999998|2337.4800000000005|
0|
|0000***4729|JOSILEI TRINDADE ...|FEMININO|  50| RJ|CEDIDO SUS LEI 8270|     ESTAVEL|            302.18|           2010.96|
0|
|0000***4287|JOANA CORREA DE S...|FEMININO|  82| PA|         APOSENTADO|     ESTAVEL|             45.99|           1384.39|
0|
|0000***3127|LARISSA DE SOUZA ...|FEMININO|  36| DF|CELETISTA EMPREGADO|     ESTAVEL|           3887.82| 8048.039999999999|
0|
|0000***9582|MARIA EUNICE DE J...|FEMININO|  76| BA|         APOSENTADO|     ESTAVEL|             50.06|            713.1|
0|
+-----------+--------------------+---------+----+---+-------------------+------------+------------------+------------------+-
-------+
only showing top 20 rows

Number of rows in CSV: 6178
```

Em seguida, armazenei o dataset no Bigquery, eu seu estado ainda bruto, para qualidade dos dados e linhagem.

```
In [4]: # Autenticação no BigQuery
        cliente_bq = bigquery.Client()

        # Defina o nome do projeto e do conjunto de dados no BigQuery
        dataset_id = "CREDITO_CONSIGNADO"

        # Armazenando os dados brutos no bigquery
        df_fe.write.format("bigquery").option("temporaryGcsBucket", "").option("writeMethod", "DIRECT").option("project", project_id).opt

        # Verificando a quantidade de registros importados para observabilidade

        query = f"SELECT COUNT(*) as total_rows FROM `halogen-antenna-426018-b0.CREDITO_CONSIGNADO.tbl_consignado_raw`"
        query_job = cliente_bq.query(query)
        result = query_job.result()
        num_rows_bq = list(result)[0].total_rows
        print(f"Number of rows in BigQuery table: {num_rows_bq}")

Number of rows in BigQuery table: 6178
```

Na sequência foi observado no dataset, CPFs duplicados, devido a anonimização. Dito isto, foram removidas as duplicidades.

```
In [5]: ##############################
        #CAMADA PRATA (Tranformação)
        ##############################

        ## Mantenha apenas a primeira ocorrência por CPF
        # Verificando duplicidade no CPF, devido a anonimização do campo para este MVP
        dataset_sem_duplicatas = df_fe.withColumn("row_num", F.row_number().over(Window.partitionBy("CPF").orderBy(F.col("cpf").desc())))
        dataset_sem_duplicatas = dataset_sem_duplicatas.filter(F.col("row_num") == 1)

        dataset_sem_duplicatas.show()

        dataset_sem_duplicatas = dataset_sem_duplicatas.filter(F.col("row_num") == 1).drop("row_num")
        df_fe = dataset_sem_duplicatas
```

Como a análise será realizada por faixa de idade, foi necessária adição das mesmas ao dataset. Ao término, armazenei o novo dataset em formato parquet ao Cloud Storage GCP, na pasta Prata.

```
# Crie colunas para cada faixa etária
df_fe = df_fe.withColumn("Faixa_10_20", F.when(df_fe["idade"].between(10, 20), 1).otherwise(0))
df_fe = df_fe.withColumn("Faixa_21_30", F.when(df_fe["idade"].between(21, 30), 1).otherwise(0))
df_fe = df_fe.withColumn("Faixa_31_40", F.when(df_fe["idade"].between(31, 40), 1).otherwise(0))
df_fe = df_fe.withColumn("Faixa_41_50", F.when(df_fe["idade"].between(41, 50), 1).otherwise(0))
df_fe = df_fe.withColumn("Faixa_51_60", F.when(df_fe["idade"].between(51, 60), 1).otherwise(0))
df_fe = df_fe.withColumn("Faixa_61_70", F.when(df_fe["idade"].between(61, 70), 1).otherwise(0))
df_fe = df_fe.withColumn("Faixa_71_80", F.when(df_fe["idade"].between(71, 80), 1).otherwise(0))
df_fe = df_fe.withColumn("Faixa_81_90", F.when(df_fe["idade"].between(81, 90), 1).otherwise(0))

df_fe.show()
```

```
+-----------+--------------------+---------+-----+---+--------------------+-------------+----------------+------------------+
|        cpf|                nome|     sexo|idade| uf| situacao_funcional|tipo_contrato|        pmt_valor|      margem_saldo|
conceder|Faixa_10_20|Faixa_21_30|Faixa_31_40|Faixa_41_50|Faixa_51_60|Faixa_61_70|Faixa_71_80|Faixa_81_90|
+-----------+--------------------+---------+-----+---+--------------------+-------------+----------------+------------------+
--------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+
|***8***3287|CARLOS ALBERTO S ...| FEMININO|   76| AM|          APOSENTADO|     ESTAVEL|1540.4199999999998|            4998.18|
0|          0|          0|          0|          0|          0|          1|          0|
|***2***7164|ARIANA MARIA DE C...| FEMININO|   37| MT|CELETISTA EMPREGADO|     ESTAVEL|           3508.71|            3184.28|
1|          0|          0|          1|          0|          0|          0|          0|
|***4***3744|ELIANE GOMES DA S...| FEMININO|   55| RJ|          APOSENTADO|     ESTAVEL|           2048.42|-3172.8800000000006|
1|          0|          0|          0|          0|          1|          0|          0|
|***6***0365|DANNIEL ROCHA DO ...|MASCULINO|   38| PI|   ATIVO PERMANENTE|     ESTAVEL|           4678.62|           10253.44|
0|          0|          0|          1|          0|          0|          0|          0|
|***8***4172|SUZANA DE MATTOS ...| FEMININO|   84| MG|BENEFICIARIO PENSAO| NAO ESTAVEL|           1584.85|            381.75|
1|          0|          0|          0|          0|          0|          0|          1|
|***9***0463|GILSON EDMAR GONC...| FEMININO|   78| PE|          APOSENTADO|     ESTAVEL|            652.84|            8514.0|
```

1.4.3. Camada Ouro (Carga)

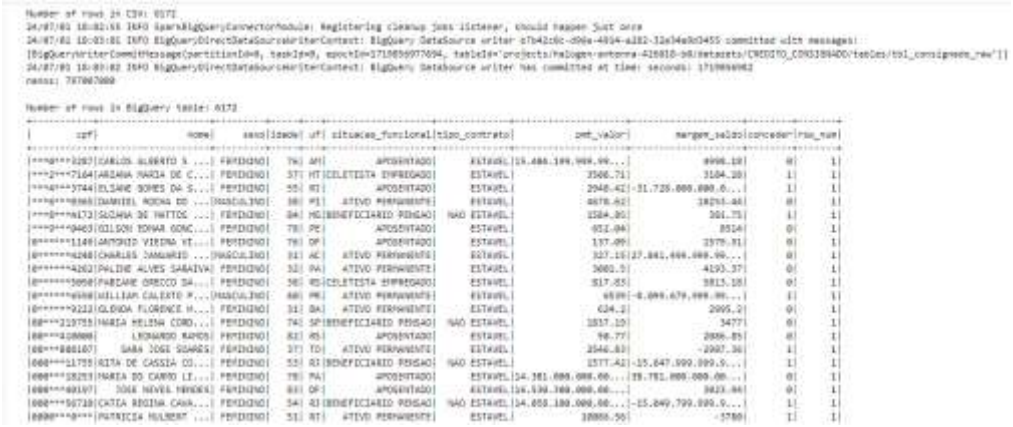
Basicamente o dataset foi agrupado com as colunas UF, Margem_Saldo, Grupo de faixas e adicionado o campo id (identity). Após isso, gravamos o dataset no bigquery como tabela fato (FLAT), para realização das análises.

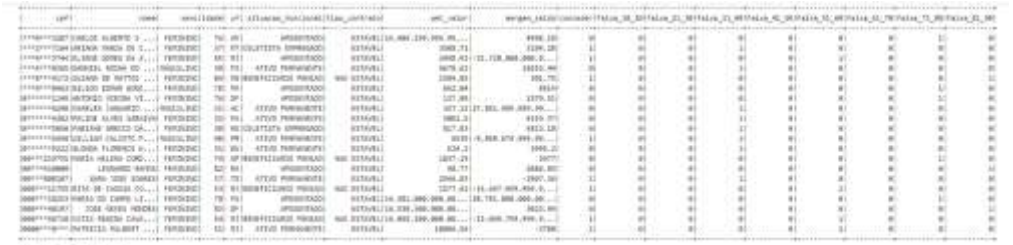
```
+---+---+------------------+----------+----------+----------+----------+----------+----------+----------+----------+
| id| uf|      Margem_Saldo|Faixa_10_20|Faixa_21_30|Faixa_31_40|Faixa_41_50|Faixa_51_60|Faixa_61_70|Faixa_71_80|Faixa_81_90|
+---+---+------------------+----------+----------+----------+----------+----------+----------+----------+----------+
|  0| MS|253197.91000000003|         1|         1|        84|         5|         4|         5|         3|         4|
|  1| CE|         1248672.97|         1|         0|       105|         7|         3|         6|        73|        93|
|  2| MG|1888138.9199999974|         0|         1|        21|       167|        34|        32|        62|       110|
|  3| DF|3392125.6299999985|         0|         8|       417|        47|        20|         6|        82|       108|
|  4| RO|          40581.68|         1|        13|        21|         1|         1|         2|         4|         0|
|  5| AM| 610785.1200000005|         1|        13|        37|         6|         0|         0|        42|        34|
|  6| MT|375235.17000000016|         0|         0|        55|         4|         4|         4|        10|        13|
|  7| SP|1064686.4700000007|         1|         2|        48|        28|        18|       142|        78|        30|
|  8| PB|          315985.21|         0|         1|         9|        59|         9|         6|        14|         9|
|  9| BA| 863103.3799999999|         0|         0|        67|        58|        10|        10|        38|        52|
| 10| SE|280744.18000000005|         0|         0|        35|        19|         1|         2|         6|         1|
| 11| RJ|1361111.9599999997|         0|         1|        15|        57|       681|       130|       120|        64|
| 12| AC| 64217.08999999999|         0|         8|        11|         1|         2|         1|         5|         3|
| 13| PR| 581894.5800000003|         0|         1|        20|        17|         3|         7|        34|        42|
| 14| AP|116248.91000000002|         0|         6|        22|         0|         1|         0|        18|         6|
| 15| RR|16245.420000000002|         1|         8|        19|         4|         1|         1|         1|         0|
| 16| TO|          154264.8|         0|         1|        36|         8|         4|         0|         2|         1|
| 17| ES|         215858.67|         0|         0|         1|         8|        27|         5|         6|         5|
| 18| ND|            1498.5|         2|         0|         0|         0|         0|         0|         0|         0|
| 19| AL|         124588.56|         0|         0|         5|        16|         4|         5|         1|         8|
+---+---+------------------+----------+----------+----------+----------+----------+----------+----------+----------+
only showing top 20 rows
```

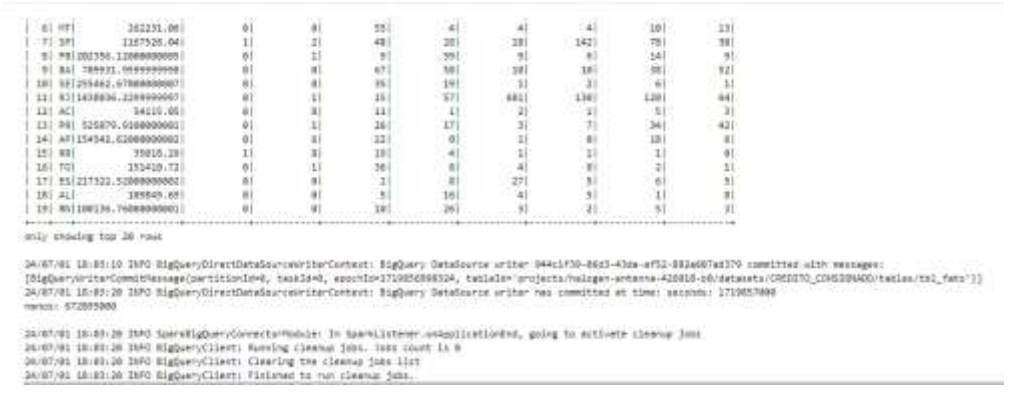
OBS.: Útilizamos o **modelo flat** para o data warehouse.

1.4.3.1. Dataproc

Selecionamos o Google DataProc como gerenciador do Spark e Hadoop, para execução de toda pipeline deste projeto. Criamos o JOB denominado Pipeline_FULL, apontando para o arquivo pyspark (pipeline.py), armazenado no bucket Bronze. Abaixo a saída de todo fluxo:







1.4.3.2. Linhagem

Como os dados não sofreram alteração, por ser uma base real já tratada, não conseguirei detalhamento sobre o tema. Deixarei o print abaixo, como forma de representação do controle no bigquery:

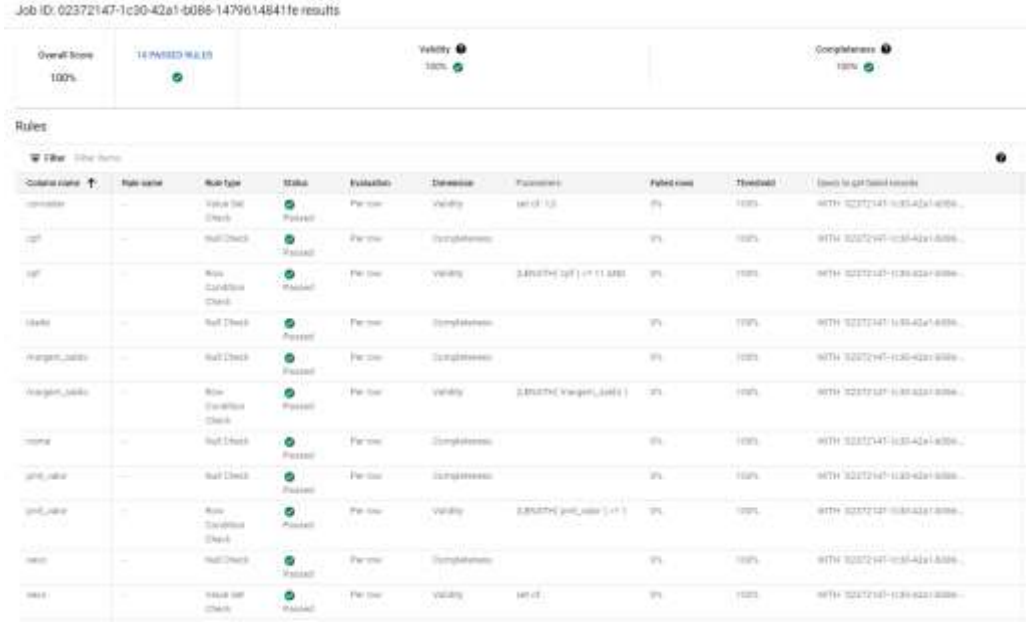
1.4.3.3. Dataplex (Data Quality)

Utilizamos o Dataplex para realização do Data Quality. Foram adicionadas regras de negócio para validação dos campos do dataset original. Importante lembrar que os dados de origem já estavam previamente tratados, por se tratar de uma base privada.

Configuração do Dataplex com as regras de negócio:

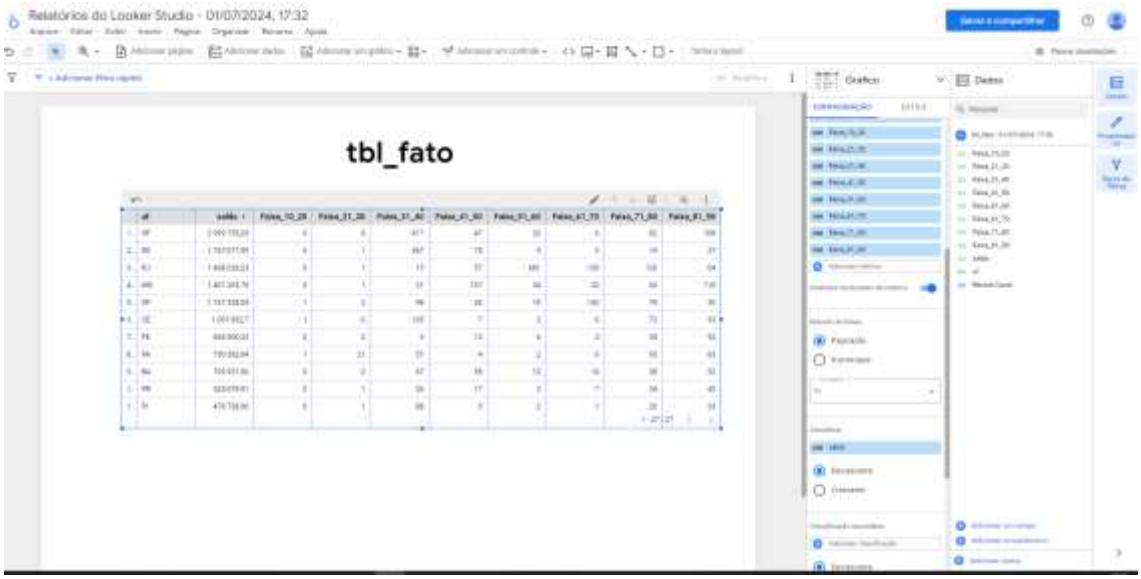
Sucesso na validação do Job:

Detalhamento dos testes realizados:



1.4.4. Looker Stúdio (Análises)

O Looker Stúdio é uma ferramenta de análise acoplada ao Bigquery.



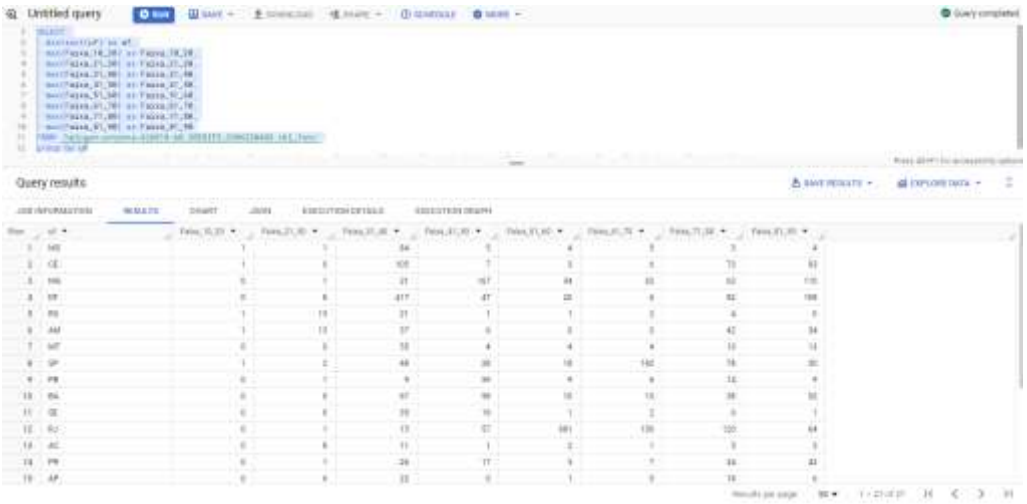
1.4.4.1. Volume de idade por UF

Neste item, precisamos entender se faz sentido segmentarmos a base pela UF e grupo de faixas de atrasos.
Baseado no gráfico abaixo, percebemos nitidamente uma aglomeração muito alta na faixa de 31_40, para minha "surpresa". Seguidos pela faixa de 81_90, 51_60 e 71_80.

Consulta:

```
SELECT
  distinct(uf) as uf,
  Margem_Saldo as saldo,
  Faixa_10_20,
  Faixa_21_30,
  Faixa_31_40,
  Faixa_41_50,
  Faixa_51_60,
  Faixa_61_70,
  Faixa_71_80,
  Faixa_81_90
FROM `halogen-antenna-426018-b0.CREDITO_CONSIGNADO.tbl_fato`
--group by uf
order by Margem_Saldo desc
```

Execução no Bigquery:



	uf	Faixa_10_20	Faixa_21_30	Faixa_31_40...	Faixa_51_60	Faixa_61_70	Faixa_71_80	Faixa_81_90
1.	RS	0	1	467	9	5	19	37
2.	DF	0	8	417	20	6	82	108
3.	GO	0	3	125	4	2	6	5
4.	CE	1	0	105	3	6	73	93
5.	PI	0	1	88	2	1	26	29
6.	MS	1	1	84	4	5	3	4
7.	MA	1	0	76	2	2	16	28
8.	BA	0	0	67	10	10	38	52
9.	PA	1	21	57	2	0	95	83
10.	MT	0	0	55	4	4	10	13
11.	SP	1	2	48	18	142	78	30
12.	AM	1	13	37	0	0	42	34
13.	TO	0	1	36	4	0	2	1
14.	SE	0	0	35	1	2	6	1
15.	SC	0	0	32	10	5	12	14
16.	PR	0	1	26	3	7	34	42
17.	AP	0	6	22	1	0	18	6
18.	RO	1	13	21	1	2	4	0
19.	MG	0	1	21	34	32	62	110
20.	RR	1	8	19	1	1	1	0
	Total geral	**8**	**90**	**1.898**	**865**	**384**	**813**	**874**

Já em relação a UF, notamos como aglomeração nas UFs: RS,DF,GO,CE para faixa 31_40:

	uf	Faixa_10_20	Faixa_21_30	Faixa_31_40...	Faixa_51_60	Faixa_61_70	Faixa_71_80	Faixa_81_90
1.	RS	0	1	467	9	5	19	37
2.	DF	0	8	417	20	6	82	108
3.	GO	0	3	125	4	2	6	5
4.	CE	1	0	105	3	6	73	93
5.	PI	0	1	88	2	1	26	29
6.	MS	1	1	84	4	5	3	4
7.	MA	1	0	76	2	2	16	28
8.	BA	0	0	67	10	10	38	52
9.	PA	1	21	57	2	0	95	83
10.	MT	0	0	55	4	4	10	13
11.	SP	1	2	48	18	142	78	30
12.	AM	1	13	37	0	0	42	34
13.	TO	0	1	36	4	0	2	1
14.	SE	0	0	35	1	2	6	1
15.	SC	0	0	32	10	5	12	14
16.	PR	0	1	26	3	7	34	42
17.	AP	0	6	22	1	0	18	6
18.	RO	1	13	21	1	2	4	0
19.	MG	0	1	21	34	32	62	110
20.	RR	1	8	19	1	1	1	0
	Total geral	**8**	**90**	**1.898**	**865**	**384**	**813**	**874**

Na faixa de 81_90, notamos interseção apenas na UF de DF:

	uf	Faixa_10_20	Faixa_21_30	Faixa_31_40	Faixa_51_60	Faixa_61_70	Faixa_71_80	Faixa_81_90...
1.	MG	0	1	21	34	32	62	110
2.	DF	0	8	417	20	6	82	108
3.	CE	1	0	105	3	6	73	93
4.	PE	0	0	9	6	3	35	92
5.	PA	1	21	57	2	0	95	83
6.	RJ	0	1	15	681	130	120	64
7.	BA	0	0	67	10	10	38	52
8.	PR	0	1	26	3	7	34	42
9.	RS	0	1	467	9	5	19	37
10.	AM	1	13	37	0	0	42	34
11.	SP	1	2	48	18	142	78	30
12.	PI	0	1	88	2	1	26	29
13.	MA	1	0	76	2	2	16	28
14.	SC	0	0	32	10	5	12	14
15.	MT	0	0	55	4	4	10	13
16.	PB	0	1	9	9	6	14	9
17.	AL	0	0	5	4	5	1	8
18.	AP	0	6	22	1	0	18	6
19.	GO	0	3	125	4	2	6	5
20.	ES	0	0	1	27	5	6	5
	Total geral	8	90	1.898	865	384	813	874

1 - 27 / 27

1.4.4.2. Volume da margem de saldo por UF

Neste item, precisamos entender se faz sentido segmentarmos a base pela UF e margem de saldo. Através do gráfico abaixo, percebemos que DF, RS, RJ, MG, SP e CE possuem um valor financeiro considerável.

Consulta:

```
SELECT
  distinct(uf) as uf,
  Margem_Saldo as saldo,
FROM `halogen-antenna-426018-b0.CREDITO_CONSIGNADO.tbl_fato`
order by Margem_Saldo desc
```

Execução no Bigquery:

Untitled query

RUN SAVE DOWNLOAD SHARE SCHEDULE MORE

```
SELECT
  distinct(uf) as uf,
  Margem_Saldo as saldo,
FROM `halogen-antenna-426018-b0.CREDITO_CONSIGNADO.tbl_fato`
--group by uf
order by Margem_Saldo desc
```

Query results

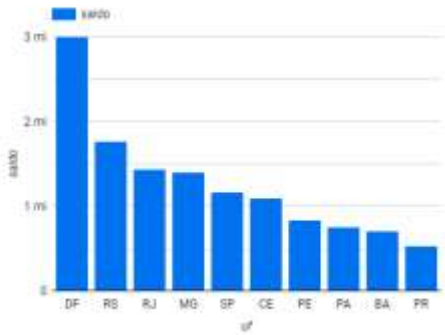
JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	uf	saldo
1	DF	2999753.249999...
2	RS	1767017.589999...
3	RJ	1438036.229999...
4	MG	1401263.780000...
5	SP	1167528.04
6	CE	1097832.699999...
7	PE	836900.2299999...
8	PA	750562.9399999...
9	BA	709931.9599999...
10	PR	525079.9100000...
11	PI	478728.9599999...
12	SC	462704.87
13	GO	438537.5600000...
14	AM	430345.5899999...
15	MA	408980.8600000...

tbl_fato

	uf	saldo
1.	DF	2.999.753,...
2.	RS	1.767.017,...
3.	RJ	1.438.036,...
4.	MG	1.401.263,...
5.	SP	1.167.528,...
6.	CE	1.097.832,7
7.	PE	836.900,23
8.	PA	750.562,94
9.	BA	709.931,96
10.	PR	525.079,91

1 - 27 / 27



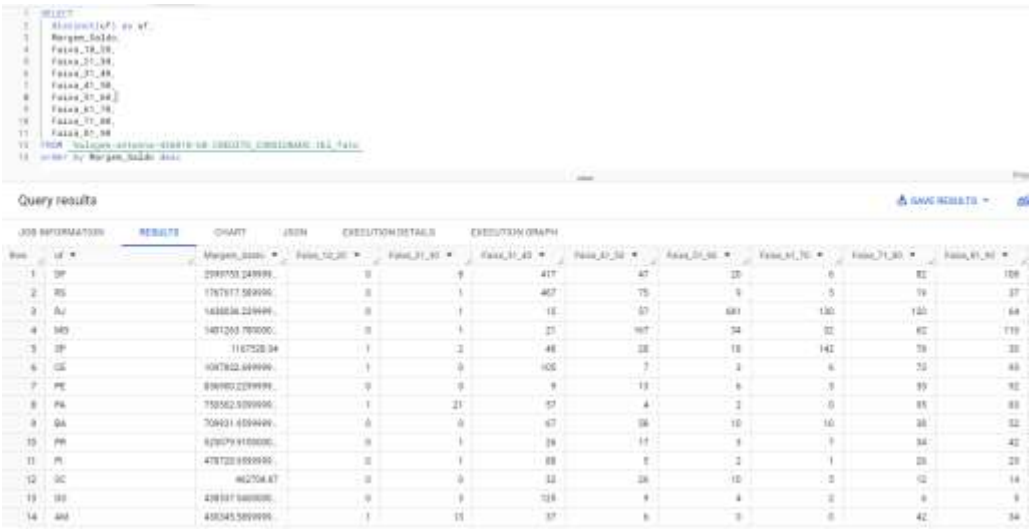
1.4.4.3. Volume da margem de saldo por UF e grupo de idade

Neste último item, precisamos entender se faz sentido segmentarmos a base pelo saldo, uf e grupo de idade. Utilizamos como referência a faixa de 31_40, devido ao alto volume encontrado no item 1.4.4.1, deste documento.

Consulta:

```
SELECT
  distinct(uf) as uf,
  Margem_Saldo as saldo,
  Faixa_10_20,
  Faixa_21_30,
  Faixa_31_40,
  Faixa_41_50,
  Faixa_51_60,
  Faixa_61_70,
  Faixa_71_80,
  Faixa_81_90
FROM `halogen-antenna-426018-b0.CREDITO_CONSIGNADO.tbl_fato`
order by Margem_Saldo desc
```

Execução no Bigquery:



Através do gráfico abaixo, percebemos que RS, DF, GO e CE como UFs relevantes.

	uf	saldo	Faixa_10_20	Faixa_21_30	Faixa_31_4...	Faixa_41_50	Faixa_51_60	Faixa_61_70	Faixa_71_80	Faixa_81_90
1	RS	1.767.017,59	0	1	467	75	9	5	19	37
2	DF	2.999.753,25	0	8	417	47	20	6	82	108
3	GO	438.537,56	0	3	125	9	4	2	6	5
4	CE	1.097.832,7	1	0	105	7	3	6	73	93
5	PI	478.728,96	0	1	88	5	2	1	28	29
6	MS	200.709,97	1	1	84	5	4	5	3	4
7	MA	408.980,86	1	0	76	5	2	2	16	28
8	BA	709.931,96	0	0	67	58	10	10	38	52
9	PA	750.562,94	1	21	57	4	2	0	95	83
10	MT	262.231,06	0	0	55	4	4	4	10	13
11	SP	1.167.528,04	1	2	48	28	18	142	78	30

1.4.5. Conclusão

1.4.5.1. Considerações

A faixa de 31 à 40 anos se destaca em termos de volume e margem de saldo, indicando um potencial interessante para oferta do crédito consignado. Confesso que fiquei muito surpreso, pois não tinha essa visão. Concentrava grande parte das ações nas faixas superiores à 50 anos.

As UF: RS,DS,GO,CE apresentam concentração significativa na faixa etária desejada e com margem de saldo favorável, tornando as áreas prioritárias para prospecção.

A análise detalhada por UF e faixa etária permite identificar oportunidades específicas para cada região, otimizando as estratégias de venda.

1.4.5.2. Recomendações

- Focar na faixa etária de 31 a 40 anos: Direcionar esforços de marketing e vendas para este público, que apresenta maior potencial de conversão.

- Priorizar as UFs RS, DF, GO e CE: Concentrar ações nestas regiões, onde existe maior concentração de clientes com perfil adequado ao crédito consignado.

- Analisar dados por UF e faixa etária: Segmentar as análises para identificar oportunidades específicas em cada região e faixa etária, personalizando as ofertas e otimizando os resultados.

- Monitorar indicadores de performance: Acompanhar métricas como taxa de aprovação, volume de concessões e inadimplência para avaliar a efetividade das estratégias e realizar ajustes quando necessário.