Projeto MVP: Análise de Avaliações de Medicamentos (Spring Egenharia da Dados PUC-Rio)

Nome: Sérgio Luiz Oliveira da Silva Filho

Matrícula: 40530010055

--- OBSERVAÇÕES IMPORTANTES ---

Distinção entre os Sprints

Sprint 1 – Projeto Inicial de Análise de Dados

No primeiro projeto desenvolvido ao longo da disciplina, o foco esteve na **análise exploratória de dados (EDA)** e na **resolução de um problema analítico específico**, utilizando um conjunto de dados previamente disponibilizado.

As principais atividades realizadas nesse sprint foram:
- Importação e leitura do dataset em ambiente local ou notebook interativo;
- Limpeza básica dos dados (tratamento de valores nulos, renomeação de colunas e ajustes de tipos);
- Análise exploratória com estatísticas descritivas e visualizações;
- Formulação de hipóteses e interpretação dos resultados obtidos.

📌 **Projeto Sprint 1:**
🔗 https://github.com/SergioOliveirasci/sergiooliveira_mvp2025-1

Esse primeiro trabalho teve caráter **exploratório e analítico**, sem a exigência de um pipeline estruturado de dados em nuvem.

Sprint 2 – Pipeline Analítico com Dados Estruturados

No segundo projeto, houve uma evolução em relação ao Sprint 1, com maior ênfase em **organização, transformação e persistência dos dados**, aproximando-se de um fluxo de engenharia de dados.

As atividades desenvolvidas incluíram:
- Estruturação mais clara do processo de ETL (Extração, Transformação e Carga);
- Padronização dos dados para facilitar análises posteriores;
- Persistência dos dados tratados em estruturas intermediárias;
- Consolidação das análises analíticas a partir dos dados transformados.

📌 **Projeto Sprint 2:**
🔗 https://github.com/SergioOliveirasci/sergiooliveira_mvp2025-2

Apesar do avanço estrutural, o projeto ainda não contemplava integralmente conceitos de **arquitetura em nuvem**, **linhagem**, **catálogo de dados** e **camadas analíticas formais**.

Diferencial do Projeto Atual (MVP 3)

O presente projeto representa uma **evolução significativa** em relação aos dois sprints anteriores, ao implementar um **pipeline completo de dados em nuvem**, utilizando a plataforma **Databricks Community Edition**, alinhado às boas práticas de **Data Engineering**, com o objetivo de analisar dados públicos sobre medicamentos, fabricantes, usos terapêuticos e avaliações de usuários.

Os principais diferenciais deste MVP em relação aos projetos anteriores são:
- Implementação de uma arquitetura em camadas (**Bronze, Silver e Gold**);
- Execução integral do pipeline em ambiente de **computação em nuvem**;
- Persistência dos dados utilizando tabelas Delta;
- Aplicação sistemática de **regras de qualidade de dados**;
- Documentação de **linhagem** e **catálogo de dados**;
- Análises analíticas baseadas em tabelas consolidadas (camada Gold);
- Discussão estruturada dos resultados com foco em tomada de decisão.

Dessa forma, este MVP não se limita à análise de dados, mas demonstra a construção de uma **solução ponta a ponta**, desde a ingestão dos dados até a geração de insights analíticos, atendendo plenamente aos requisitos propostos para o desenvolvimento do MVP da disciplina.

Fonte dos Dados e Licenciamento

O conjunto de dados utilizado neste projeto foi obtido a partir da plataforma Kaggle, um repositório amplamente utilizado para fins educacionais, acadêmicos e de pesquisa em ciência de dados.

De acordo com as informações disponibilizadas na própria plataforma, o dataset possui licença de uso público (open data), sendo permitido seu uso para fins acadêmicos e não comerciais. Dessa forma, não há restrições legais ou éticas para a utilização dos dados neste MVP, uma vez que não contém informações sensíveis ou dados pessoais identificáveis.

1. Objetivo do Projeto

O objetivo deste MVP é **desenvolver e demonstrar um pipeline completo de dados em nuvem**, utilizando a plataforma **Databricks Community Edition**, para analisar dados públicos de medicamentos e identificar **padrões de aceitação dos usuários** com base em avaliações, fabricantes, usos terapêuticos e efeitos colaterais.

Busca-se transformar dados brutos em **informações analíticas confiáveis**, aplicando práticas de **engenharia de dados**, como ingestão, transformação, persistência, controle de qualidade e modelagem analítica, culminando na geração de **insights que possam apoiar análises exploratórias e tomada de decisão** no contexto da área da saúde.

Além do aspecto analítico, este projeto tem como objetivo demonstrar a aplicação prática dos conceitos abordados na disciplina, incluindo:
- Arquitetura em camadas (Bronze, Silver e Gold);
- Processos de ETL em ambiente de nuvem;
- Avaliação da qualidade dos dados;
- Estruturação e interpretação de análises analíticas.

2. Perguntas de Negócio

Com base nos dados analisados, o projeto busca responder às seguintes perguntas de negócio:

1. **Quais fabricantes possuem os medicamentos com melhores avaliações dos usuários?**
   Esta análise permite identificar fabricantes associados a maiores percentuais de avaliações positivas, indicando possível maior aceitação ou qualidade percebida.

2. **Existe relação entre o uso terapêutico dos medicamentos e a avaliação positiva recebida?**
   Busca-se compreender se determinados tipos de uso terapêutico estão associados a níveis mais elevados de satisfação dos usuários.

3. **Medicamentos com maior quantidade de efeitos colaterais tendem a receber piores avaliações?**
   Esta pergunta investiga a possível correlação entre a ocorrência de efeitos colaterais e a percepção negativa dos usuários.

4. **Quais medicamentos apresentam os maiores percentuais de avaliações excelentes?**
   O objetivo é identificar medicamentos com maior destaque em termos de aceitação geral.

5. **Há concentração de avaliações positivas em um número reduzido de fabricantes?**
   Essa análise avalia se o mercado apresenta concentração de avaliações excelentes em poucos fabricantes ou se a aceitação está distribuída de forma mais homogênea.

3. Ingestão dos Dados – Camada Bronze

Nesta etapa do projeto é realizada a **ingestão dos dados brutos**, que compõem a **Camada Bronze** da arquitetura adotada.

A Camada Bronze tem como objetivo armazenar os dados **em seu estado mais próximo da forma original**, preservando a estrutura e o conteúdo conforme disponibilizados na fonte de origem. Nesta fase, **não são aplicadas transformações ou tratamentos analíticos**, garantindo a rastreabilidade e a possibilidade de reprocessamento dos dados, caso necessário.

Os dados utilizados neste projeto já se encontram carregados no ambiente Databricks e são acessados por meio de uma **tabela Delta**, criada a partir do dataset original. Essa abordagem permite:

- Persistência eficiente dos dados em nuvem;
- Reutilização dos dados por diferentes etapas do pipeline;
- Garantia de consistência entre execuções.

O código a seguir realiza a leitura da tabela correspondente aos dados brutos e exibe seu conteúdo inicial, permitindo a validação da estrutura, dos tipos de dados e das colunas disponíveis antes do início das transformações subsequentes.

8

```
# Camada BRONZE - Dados Brutos
df_bronze = spark.table("workspace.default.medicine_details")
display(df_bronze)
```

df_bronze: pyspark.sql.connect.dataframe.DataFrame = [Medicine Name: string, Composition: string ... 7 more fields]

Table

	Medicine Name	Composition	Uses
1	Avastin 400mg Injection	Bevacizumab (400mg)	Cancer of colon and rectum Non-small cell lung cancer Kidney cancer Bre
2	Augmentin 625 Duo Tablet	Amoxycillin (500mg) + Clavulanic Acid (125mg)	Treatment of Bacterial infections
3	Azithral 500 Tablet	Azithromycin (500mg)	Treatment of Bacterial infections
4	Ascoril LS Syrup	Ambroxol (30mg/5ml) + Levosalbutamol (1mg/5ml) + Guaifenesin (50mg/5ml)	Treatment of Cough with mucus
5	Aciloc 150 Tablet	Ranitidine (150mg)	Treatment of Gastroesophageal reflux disease (Acid reflux)Treatment of Pe
6	Allegra 120mg Tablet	Fexofenadine (120mg)	Treatment of Sneezing and runny nose due to allergiesTreatment of Allerg
7	Avil 25 Tablet	Pheniramine (25mg)	Treatment of Allergic conditionsTreatment of Respiratory disease with
8	Aricep 5 Tablet	Donepezil (5mg)	Alzheimer's disease
9	Amoxyclav 625 Tablet	Amoxycillin (500mg) + Clavulanic Acid (125mg)	Treatment of Bacterial infections
10	Atarax 25mg Tablet	Hydroxyzine (25mg)	Treatment of AnxietyTreatment of Skin conditions with inflammation & itc
11	Azee 500 Tablet	Azithromycin (500mg)	Treatment of Bacterial infections
12	Anovate Cream	Phenylephrine (0.10% w/w) + Beclometasone (0.025% w/w) + Lidocaine (2.50% w/w)	Treatment of Piles
13	Allegra-M Tablet	Montelukast (10mg) + Fexofenadine (120mg)	Treatment of Sneezing and runny nose due to allergies
14	Ascoril D Plus Syrup Sugar Free	Phenylephrine (5mg) + Chlorpheniramine Maleate (2mg) + Dextromethorphan Hydrobromide (10mg)	Treatment of Dry cough
15			

5,368+ rows | Truncated data

Validação Inicial dos Dados Brutos

A visualização dos dados confirma que a ingestão na **Camada Bronze** foi realizada com sucesso. Observa-se que o conjunto de dados contém informações detalhadas sobre medicamentos, incluindo nome, composição, uso terapêutico e avaliações dos usuários.

O volume de registros e a estrutura das colunas estão de acordo com o esperado, indicando que os dados foram carregados corretamente e sem perdas aparentes. Nesta etapa, não foram realizadas transformações, mantendo os dados em seu formato original, conforme o princípio da Camada Bronze.

Essa validação inicial garante uma base confiável para a próxima etapa do pipeline, na qual serão aplicadas operações de limpeza, padronização e tratamento de qualidade na **Camada Silver**.

4. Transformação dos Dados (Camada Silver)

A Camada Silver representa a etapa intermediária do pipeline de dados, na qual os dados brutos provenientes da Camada Bronze passam por **processos de limpeza, padronização e organização**, com o objetivo de torná-los **consistentes, confiáveis e adequados para análises analíticas**.

Nesta fase, são aplicadas transformações controladas que **não alteram o significado dos dados**, mas corrigem problemas comuns encontrados em bases reais, como inconsistências de formatação, variações de texto e nomenclaturas inadequadas para análise.

As principais atividades realizadas nesta etapa incluem:

- **Padronização de campos textuais**, por meio da remoção de espaços em branco e normalização de letras (ex.: uso de minúsculas);
- **Renomeação de colunas**, adequando os nomes para um padrão técnico consistente, facilitando consultas SQL e manipulação analítica;
- **Preparação dos dados para análise**, garantindo que as colunas estejam semanticamente claras e prontas para agregações, filtros e métricas posteriores.

Essa camada desempenha um papel fundamental na **garantia da qualidade dos dados**, servindo como base confiável para a construção da Camada Gold, onde serão realizadas as análises finais e a geração de insights.

Ao final deste processo, os dados permanecem com a mesma granularidade do dataset original, porém organizados de forma padronizada, assegurando **rastreabilidade, reprodutibilidade e clareza analítica** ao pipeline desenvolvido.

12

```
from pyspark.sql.functions import col, lower, trim, when
from pyspark.sql.types import DoubleType

df_silver = (
    df_bronze
    # Limpeza/padronização de texto (mantendo significado)
    .withColumn("Medicine Name", trim(col("Medicine Name")))
    .withColumn("Manufacturer", trim(col("Manufacturer")))
    .withColumn("Composition", trim(col("Composition")))
    .withColumn("Uses", when(col("Uses").isNotNull(), lower(trim(col("Uses")))).otherwise(col("Uses")))
    .withColumn("Side_effects", when(col("Side_effects").isNotNull(), lower(trim(col("Side_effects")))).otherwise(col("Side_effects")))

    # Renomeação padronizada para análise e SQL
    .withColumnRenamed("Medicine Name", "Medicine_Name")
    .withColumnRenamed("Image URL", "Image_URL")
    .withColumnRenamed("Excellent Review %", "Excellent_Review")
    .withColumnRenamed("Average Review %", "Average_Review")
    .withColumnRenamed("Poor Review %", "Poor_Review")

    # Garantia de tipos numéricos para métricas (caso tenham vindo como string)
    .withColumn("Excellent_Review", col("Excellent_Review").cast(DoubleType()))
    .withColumn("Average_Review", col("Average_Review").cast(DoubleType()))
    .withColumn("Poor_Review", col("Poor_Review").cast(DoubleType()))
)

(
    df_silver.write
    .mode("overwrite")
    .option("overwriteSchema", "true")
    .saveAsTable("workspace.default.medicine_details_silver")
)

display(df_silver)
print("Linhas Bronze:", df_bronze.count())
print("Linhas Silver:", df_silver.count())
```

df_silver: pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: string, Composition: string ... 7 more fields]

Table

	Medicine_Name	Composition	Uses
1	Avastin 400mg Injection	Bevacizumab (400mg)	cancer of colon and rectum non-small cell lung cancer kidney cancer brai
2	Augmentin 625 Duo Tablet	Amoxycillin (500mg) + Clavulanic Acid (125mg)	treatment of bacterial infections
3	Azithral 500 Tablet	Azithromycin (500mg)	treatment of bacterial infections
4	Ascoril LS Syrup	Ambroxol (30mg/5ml) + Levosalbutamol (1mg/5ml) + Guaifenesin (50mg/5ml)	treatment of cough with mucus
5	Aciloc 150 Tablet	Ranitidine (150mg)	treatment of gastroesophageal reflux disease (acid reflux)treatment of pe
6	Allegra 120mg Tablet	Fexofenadine (120mg)	treatment of sneezing and runny nose due to allergiestreatment of allergi
7	Avil 25 Tablet	Pheniramine (25mg)	treatment of allergic conditionstreatment of respiratory disease with e
8	Aricep 5 Tablet	Donepezil (5mg)	alzheimer's disease
9	Amoxyclav 625 Tablet	Amoxycillin (500mg) + Clavulanic Acid (125mg)	treatment of bacterial infections
10	Atarax 25mg Tablet	Hydroxyzine (25mg)	treatment of anxietytreatment of skin conditions with inflammation & itch
11	Azee 500 Tablet	Azithromycin (500mg)	treatment of bacterial infections
12	Anovate Cream	Phenylephrine (0.10% w/w) + Beclometasone (0.025% w/w) + Lidocaine (2.50% w/w)	treatment of piles
13	Allegra-M Tablet	Montelukast (10mg) + Fexofenadine (120mg)	treatment of sneezing and runny nose due to allergies
14	Ascoril D Plus Syrup Sugar Free	Phenylephrine (5mg) + Chlorpheniramine Maleate (2mg) + Dextromethorphan Hydrobromide (10mg)	treatment of dry cough
15			

5,287+ rows | Truncated data

```
Linhas Bronze: 11825
Linhas Silver: 11825
```

Validação Pós-Transformação (Camada Silver)

A execução da etapa Silver confirma que os dados passaram por **padronização e limpeza básica** com sucesso, mantendo a granularidade do dataset original.
Observa-se que:

- Os campos textuais relevantes foram **normalizados** (remoção de espaços excedentes e padronização de caixa em `Uses` e `Side_effects`), reduzindo inconsistências de escrita e facilitando análises por agrupamento.
- As colunas foram **renomeadas para um padrão consistente** (snake_case), melhorando a legibilidade e simplificando consultas SQL.
- Os percentuais de avaliação (`Excellent_Review`, `Average_Review`, `Poor_Review`) foram preparados para uso analítico, permitindo agregações e comparações de forma confiável.

A Camada Silver passa a atuar como a **base confiável** para a etapa seguinte (Camada Gold), onde serão realizadas agregações, métricas e consultas orientadas às perguntas de negócio.

Conferência de integridade: foi realizada a comparação de quantidade de registros entre Bronze e Silver para garantir que as transformações não causaram perdas indevidas de dados.

5. Camada Analítica (Gold)

A Camada Gold representa a etapa final do pipeline, na qual os dados já tratados e padronizados na Camada Silver são **agregados e estruturados** para suportar análises analíticas e responder diretamente às **perguntas de negócio** definidas no início do projeto.

Nesta camada, são criadas tabelas orientadas a consumo analítico (por exemplo, **métricas por fabricante, uso terapêutico ou medicamento**), com o objetivo de:

- Reduzir a complexidade das consultas finais (deixando-as mais simples e rastreáveis);
- Melhorar desempenho na exploração analítica (dados já consolidados);
- Garantir consistência das métricas, evitando múltiplas versões de um mesmo cálculo.

Neste primeiro artefato da Camada Gold, é construída uma tabela agregada **por fabricante**, calculando métricas de avaliação média dos usuários, como:

- média de avaliações excelentes (*Excellent Review*);
- média de avaliações médias (*Average Review*).

O resultado é persistido como tabela Delta na camada Gold e utilizado nas análises subsequentes.

17

```
from pyspark.sql.functions import avg, round, col

df_gold = (
    spark.table("workspace.default.medicine_details_silver")
    .groupBy("Manufacturer")
    .agg(
        round(avg("Excellent_Review"), 2).alias("avg_excellent_review"),
        round(avg("Average_Review"), 2).alias("avg_average_review"),
        round(avg("Poor_Review"), 2).alias("avg_poor_review")
    )
    .orderBy(col("avg_excellent_review").desc())
)

(
    df_gold.write
    .mode("overwrite")
    .option("overwriteSchema", "true")
    .saveAsTable("workspace.default.medicine_manufacturer_gold")
)

display(df_gold)
```

df_gold: pyspark.sql.connect.dataframe.DataFrame = [Manufacturer: string, avg_excellent_review: double ... 2 more fields]

Table

	Manufacturer	1.2 avg_excellent_review	1.2 avg_average_review	1.2 avg_poor_review
1	Indian Immunologicals Ltd	100	0	0
2	Vaishali Pharma Ltd	100	0	0
3	Matteo Healthcare Pvt Ltd	100	0	0
4	Gamete Pharma Pvt Ltd	100	0	0
5	Bio-Med Pvt Ltd	100	0	0
6	ADZO Lifesciences Pvt Ltd	100	0	0
7	Ampra Pharmaceuticals	100	0	0
8	Cure Quick Pharmaceuticals	100	0	0
9	Dermcure's Pharma	100	0	0
10	Holy Lifesciences	100	0	0
11	Sigman Wellness	100	0	0
12	Zunison Healthcare	100	0	0
13	Avirav Sciences Pvt Ltd	100	0	0
14	Depsons Healthcare	100	0	0
15	Coxswain Healthcare	100	0	0

759 rows

Validação e Interpretação Inicial (Camada Gold)

A tabela gerada na Camada Gold consolida os dados **por fabricante**, apresentando as métricas médias de avaliação dos usuários.
Observa-se que o resultado contém aproximadamente **centenas de fabricantes (≈ 759 linhas)**, indicando que os dados foram agregados corretamente a partir da Camada Silver.

Nesta etapa, o pipeline transforma registros individuais de medicamentos em uma visão analítica resumida, permitindo:

- comparar fabricantes com base em indicadores de aceitação dos usuários;
- identificar fabricantes com maior média de avaliações excelentes;
- apoiar análises de concentração de avaliações positivas (Top N fabricantes).

Essas métricas serão utilizadas nas seções seguintes para responder às perguntas de negócio, com consultas adicionais e discussão dos resultados.

Observação: A ordenação por `avg_excellent_review` facilita a visualização imediata dos fabricantes com melhor desempenho médio, servindo como base para o ranking da Pergunta 1.

6. Análise da Qualidade dos Dados

Nesta etapa é realizada a análise da qualidade dos dados consolidados na **Camada Silver**, com o objetivo de garantir que os atributos estejam consistentes, confiáveis e adequados para a construção das análises analíticas na Camada Gold.

A verificação de qualidade vai além da simples identificação de valores nulos, contemplando também regras básicas de consistência e coerência dos dados, tais como:

- presença de valores nulos e campos textuais vazios;
- verificação de faixas válidas para os percentuais de avaliação dos usuários;
- validação da consistência da soma dos percentuais de avaliação;
- identificação de possíveis registros duplicados com base em uma chave lógica.

Essas análises fornecem evidências de controle de qualidade ao pipeline e aumentam a confiabilidade das métricas e insights produzidos nas etapas seguintes do projeto.

21

```
from pyspark.sql import functions as F

df_quality = spark.table("workspace.default.medicine_details_silver")

# ========================
# 1) Valores Nulos por Coluna
# ========================
nulls_df = df_quality.select([
    F.sum(F.col(c).isNull().cast("int")).alias(c)
    for c in df_quality.columns
])

display(nulls_df)

# ========================
# 2) Campos Vazios (strings)
# ========================
string_cols = [name for name, dtype in df_quality.dtypes if dtype == "string"]

empties_df = df_quality.select([
    F.sum((F.trim(F.col(c)) == "").cast("int")).alias(c)
    for c in string_cols
])

display(empties_df)

# ========================
# 3) Percentuais fora do intervalo [0, 100]
# ========================
pct_cols = ["Excellent_Review", "Average_Review", "Poor_Review"]

out_of_range_df = df_quality.select(
    *[
        F.sum((F.col(c) < 0).cast("int")).alias(f"{c}_below_0")
        for c in pct_cols
    ],
    *[
        F.sum((F.col(c) > 100).cast("int")).alias(f"{c}_above_100")
        for c in pct_cols
    ]
)

display(out_of_range_df)

# ========================
# 4) Consistência da soma dos percentuais (~100)
#    Tolerância: 99 a 101
# ========================
df_sum_check = df_quality.withColumn(
    "review_sum",
    F.col("Excellent_Review") + F.col("Average_Review") + F.col("Poor_Review")
)

total_rows = df_quality.count()
inconsistent_sum = df_sum_check.filter(
    (F.col("review_sum") < 99) | (F.col("review_sum") > 101)
).count()

display(
    spark.createDataFrame(
        [(total_rows, inconsistent_sum)],
        ["total_registros", "registros_com_soma_inconsistente"]
    )
)

# ========================
# 5) Duplicidade por chave lógica
#    (Medicamento + Fabricante)
# ========================
duplicates_df = (
    df_quality
    .groupBy("Medicine_Name", "Manufacturer")
    .count()
    .filter(F.col("count") > 1)
    .orderBy(F.col("count").desc())
)

display(duplicates_df)
```

```
> df_quality:  pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: string, Composition: string ... 7 more fields]
> df_sum_check:  pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: string, Composition: string ... 8 more fields]
> duplicates_df:  pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: string, Manufacturer: string ... 1 more field]
> empties_df:  pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: long, Composition: long ... 4 more fields]
> nulls_df:  pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: long, Composition: long ... 7 more fields]
> out_of_range_df:  pyspark.sql.connect.dataframe.DataFrame = [Excellent_Review_below_0: long, Average_Review_below_0: long ... 4 more fields]
```

Table

	Medicine_Name	Composition	Uses	Side_effects	Image_URL	Manufacturer	Excellent_Review	Average_Review	Poor_Review
1	0	0	0	0	0	0	0	0	0

1 row

Table

	Medicine_Name	Composition	Uses	Side_effects	Image_URL	Manufacturer
1	0	0	0	0	0	0

1 row

Table

	Excellent_Review_below_0	Average_Review_below_0	Poor_Review_below_0	Excellent_Review_above_100	Average_Review_above_100	Poor_Review_above_100
1	0	0	0	0	0	0

1 row

Table

	total_registros	registros_com_soma_inconsistente
1	11825	0

1 row

Table

	Medicine_Name	Manufacturer	count
1	Lulifin Cream	Sun Pharmaceutical Industries Ltd	4
2	Melgain Lotion	Zydus Cadila	3
3	Episert Cream	KLM Laboratories Pvt Ltd	3
4	Lulilok Cream	Oaknet Healthcare Pvt Ltd	3
5	L-Sys Cream	Systopic Laboratories Pvt Ltd	3
6	Lulibet Cream	Intas Pharmaceuticals Ltd	3
7	Nebistar 5 Tablet	Lupin Ltd	3
8	Xarelto 20mg Tablet	Bayer Zydus Pharma Pvt Ltd	3
9	Nizral Cream	Janssen Pharmaceuticals	3
10	Lofatin Cream	Abbott	3
11	Hexidine Mouth Wash	Icpa Health Products Ltd	3
12	Tcris Cream	Brinton Pharmaceuticals Pvt Ltd	3
13	Ebernet Cream	Dr Reddy's Laboratories Ltd	3
14	Sertaspor Cream	Intas Pharmaceuticals Ltd	3
15	Livoluk Oral Solution	Panacea Biotec Ltd	3

299 rows

Discussão dos Resultados da Qualidade dos Dados

A análise de qualidade realizada permite avaliar se os dados estão aptos para consumo analítico na Camada Gold.

- A verificação de **valores nulos** e **campos vazios** indica a completude das informações disponíveis.
- A validação das **faixas dos percentuais de avaliação (0–100)** assegura coerência numérica para o cálculo de métricas agregadas.
- A checagem da **soma dos percentuais de avaliação** atua como uma regra de consistência, uma vez que as avaliações representam distribuições percentuais.
- A identificação de **registros duplicados** com base na chave lógica (Medicamento + Fabricante) evita distorções nas análises, especialmente em rankings e médias por grupo.

No contexto deste MVP, os resultados obtidos são registrados como evidência do controle de qualidade do pipeline de dados. Caso fossem identificadas inconsistências críticas, estas poderiam ser tratadas na Camada Silver antes da geração das tabelas analíticas finais.

7. Análises e Respostas às Perguntas de Negócio

Nesta etapa final do MVP, os dados consolidados na **Camada Gold** são utilizados para responder, de forma objetiva e rastreável, às **perguntas de negócio** definidas no início do projeto.

As análises a seguir são organizadas por pergunta, sempre seguindo o mesmo padrão:

1. **Objetivo da pergunta** (o que será avaliado);
2. **Consulta analítica** (SQL/PySpark sobre tabelas Gold ou Silver quando necessário);
3. **Resultado** (saída apresentada no Databricks);
4. **Interpretação** (discussão do que os números indicam e como respondem à pergunta).

Esse formato garante reprodutibilidade, clareza e alinhamento com as boas práticas de engenharia de dados aplicadas ao contexto analítico.

7.1 Pergunta 1 — Quais fabricantes possuem os medicamentos com melhores avaliações dos usuários?

Objetivo: identificar fabricantes com maior aceitação média, utilizando como métrica o percentual médio de avaliações excelentes (`Excellent_Review`) e, como apoio, o percentual médio de avaliações médias (`Average_Review`), com base na tabela agregada da Camada Gold.

25

```
# 7.1 - Ranking de fabricantes por média de avaliações excelentes (Camada Gold)

from pyspark.sql import functions as F

df_manu_gold = spark.table("workspace.default.medicine_manufacturer_gold")

df_q1 = (
    df_manu_gold
    .select(
        "Manufacturer",
        F.col("avg_excellent_review"),
        F.col("avg_average_review")
    )
    .orderBy(F.col("avg_excellent_review").desc())
)

display(df_q1)
```

```
df_manu_gold: pyspark.sql.connect.dataframe.DataFrame = [Manufacturer: string, avg_excellent_review: double ... 2 more fields]
df_q1: pyspark.sql.connect.dataframe.DataFrame = [Manufacturer: string, avg_excellent_review: double ... 1 more field]
```

Table

	Manufacturer	avg_excellent_review	avg_average_review
1	Onerous Pharma	100	0
2	Vaishali Pharma Ltd	100	0
3	Gamete Pharma Pvt Ltd	100	0
4	Lividus Pharmaceuticals Pvt Ltd	100	0
5	Shilpa Medicare Ltd	100	0
6	Aeobury Healthcare Pvt Ltd	100	0
7	Zodley Pharmaceuticals Pvt Ltd	100	0
8	Coxswain Healthcare	100	0
9	Prosper Channel Lifescience India Pvt L...	100	0
10	Domagk Smith Labs Pvt Ltd	100	0
11	Clementia Pharmaceuticals Pvt Ltd	100	0
12	Sigman Wellness	100	0
13	Beulah Biomedics Ltd	100	0
14	Dr Cure Pharmaceuticals India Pvt Ltd	100	0
15	Matteo Healthcare Pvt Ltd	100	0

759 rows

Comentário após o resultado (7.1):
Os resultados indicam diversos fabricantes com média de avaliações excelentes igual a 100%. Esse comportamento sugere que, para esses fabricantes, os medicamentos disponíveis na base receberam exclusivamente avaliações classificadas como excelentes.

Entretanto, é importante destacar que esse resultado pode estar associado a **baixo número de medicamentos ou avaliações por fabricante**, o que limita a comparação direta entre eles. Assim, a análise deve ser interpretada de forma descritiva, indicando percepção positiva dos usuários, mas sem permitir inferência conclusiva sobre superioridade absoluta entre fabricantes.

7.2 Pergunta 2 — Existe relação entre o uso terapêutico e a avaliação positiva dos medicamentos?

Objetivo: avaliar se determinadas categorias de **uso terapêutico** (`Uses`) estão associadas a maiores médias de avaliações excelentes.
Como `Uses` é um campo textual (podendo conter mais de um uso), nesta análise será utilizada uma abordagem por texto padronizado presente na Camada Silver.

28

```
# 7.2 - Média de avaliações por "Uses" (Camada Silver)
# Observação: esta análise usa Silver porque o campo "Uses" é textual e pode exigir padronização.

from pyspark.sql import functions as F

df_silver = spark.table("workspace.default.medicine_details_silver")

df_q2 = (
    df_silver
    .groupBy("Uses")
    .agg(
        F.round(F.avg("Excellent_Review"), 2).alias("avg_excellent_review"),
        F.count("*").alias("qtd_registros")
    )
    .orderBy(F.col("avg_excellent_review").desc(), F.col("qtd_registros").desc())
)

display(df_q2)
```

```
df_q2: pyspark.sql.connect.dataframe.DataFrame = [Uses: string, avg_excellent_review: double ... 1 more field]
df_silver: pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: string, Composition: string ... 7 more fields]
```

Table

	Uses	avg_excellent_review	qtd_registros
1	treatment of nauseatreatment of vomitingtreatment of diarrhea	100	1
2	closure of surgical incisions	100	1
3	blood cancer (chronic myeloid leukaemia)	100	1
4	non-small cell lung cancer breast cancer pancreatic cancer urinary bladder cancer ovarian cancer	100	1
5	fibromyalgia depression	100	1
6	soft tissue sarcoma	100	1
7	calcium deficiency	100	1
8	moderate to severe pain	100	1
9	severe hypoglycemia	92	1
10	respiratory tract disorders associated with viscid mucus	89	1
11	bacterial infections of external ear fungal skin infections	86.5	2
12	prevention of harmful effects of methotrexate	86	2
13	breast cancer stomach cancer	83.5	2
14	organophosphate poisoning	83	1
15	reversing the effects of certain drugs used during surgery	83	1

712 rows

Comentário após o resultado (7.2):
A análise mostra que diversos usos terapêuticos apresentam média de avaliações excelentes igual a 100%. Esse comportamento sugere alta aceitação dos medicamentos associados a esses usos.

No entanto, o campo `Uses` é textual e pode representar combinações de múltas indicações, o que gera grande granularidade e possíveis amostras pequenas. Por esse motivo, a variável `qtd_registros` deve ser considerada na interpretação, pois usos com poucos registros podem apresentar médias extremas sem representar uma tendência geral do conjunto de dados.

7.3 Pergunta 3 — Medicamentos com mais efeitos colaterais tendem a receber piores avaliações?

Objetivo: criar uma métrica derivada de quantidade de efeitos colaterais com base no texto de `Side_effects` e comparar a média de avaliações excelentes conforme essa quantidade aumenta.

31

```
# 7.3 - Relação entre quantidade de efeitos colaterais e avaliações
from pyspark.sql import functions as F

df_silver = spark.table("workspace.default.medicine_details_silver")

# Métrica simples: número aproximado de efeitos colaterais = quantidade de itens separados por vírgula
df_side = df_silver.withColumn(
    "num_side_effects",
    F.when(F.col("Side_effects").isNull() | (F.trim(F.col("Side_effects")) == ""), F.lit(0))
     .otherwise(F.size(F.split(F.col("Side_effects"), ",")))
)

df_q3 = (
    df_side
    .groupBy("num_side_effects")
    .agg(
        F.round(F.avg("Excellent_Review"), 2).alias("avg_excellent_review"),
        F.round(F.avg("Poor_Review"), 2).alias("avg_poor_review"),
        F.count("*").alias("qtd_registros")
    )
    .orderBy(F.col("num_side_effects").asc())
)

display(df_q3)
```

df_q3: pyspark.sql.connect.dataframe.DataFrame = [num_side_effects: integer, avg_excellent_review: double ... 2 more fields]
df_side: pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: string, Composition: string ... 8 more fields]
df_silver: pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: string, Composition: string ... 7 more fields]

Table

	num_side_effects	avg_excellent_review	avg_poor_review	qtd_registros
1	1	38.52	25.73	11825

1 row

Comentário após o resultado (7.3):
O resultado indica que a maioria dos registros apresenta uma única faixa de quantidade de efeitos colaterais, o que impede a comparação entre diferentes níveis de efeitos adversos.

Dessa forma, nesta base de dados, não foi possível identificar uma relação clara entre o aumento do número de efeitos colaterais e a variação das avaliações. A análise permanece descritiva e evidencia uma limitação do dataset para esse tipo de inferência, que poderia ser explorada em trabalhos futuros com maior diversidade de registros.

7.4 Pergunta 4 — Quais medicamentos apresentam maior percentual de avaliações excelentes?

Objetivo: identificar medicamentos com maior valor de `Excellent_Review` e apresentar um ranking com base na Camada Silver.

34

```
# 7.4 - Ranking de medicamentos por Excellent_Review (Camada Silver)

from pyspark.sql import functions as F

df_silver = spark.table("workspace.default.medicine_details_silver")

df_q4 = (
    df_silver
    .select(
        "Medicine_Name",
        "Manufacturer",
        "Excellent_Review",
        "Average_Review",
        "Poor_Review"
    )
    .orderBy(F.col("Excellent_Review").desc())
)

display(df_q4)
```

df_q4: pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: string, Manufacturer: string ... 3 more fields]
df_silver: pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: string, Composition: string ... 7 more fields]

Table

	Medicine_Name	Manufacturer	1.2 Excellent_Review	1.2 Average_Review	1.2 Poor_Review
1	Intalith 300 Tablet	Intas Pharmaceuticals Ltd	100	0	0
2	Enzigest 25000 Capsule	Wallace Pharmaceuticals Pvt Ltd	100	0	0
3	Domtac-SR Capsule	Rapross Pharmaceuticals Pvt Ltd	100	0	0
4	Aminogen Eye Drop	Jawa Pharmaceuticals Pvt Ltd	100	0	0
5	Syntran SB Capsule	Glenmark Pharmaceuticals Ltd	100	0	0
6	Anbid 500 Tablet	AN Pharmaceuticals Pvt Ltd	100	0	0
7	Rimoflo-T Ophthalmic Solution	Ipca Laboratories Ltd	100	0	0
8	Solvin LS Syrup	Ipca Laboratories Ltd	100	0	0
9	Strozina Plus Tablet	Arinna Lifescience Pvt Ltd	100	0	0
10	Propess Pessaries	Ferring Pharmaceuticals	100	0	0
11	Mupidac Cream	Cadila Pharmaceuticals Ltd	100	0	0
12	Starvog GM 1 Tablet SR	Merck Ltd	100	0	0
13	Brigrel Tablet	Micro Labs Ltd	100	0	0
14	Istaphase MG 2 Tablet PR	Hetero Drugs Ltd	100	0	0
15	Itgo 50DT Tablet	Salsim Pharma Pvt Ltd	100	0	0

10,000+ rows | Truncated data

Comentário após o resultado (7.4):
O ranking apresenta diversos medicamentos com 100% de avaliações classificadas como excelentes. Esse resultado indica percepção extremamente positiva dos usuários para esses produtos específicos.

Entretanto, a ausência de avaliações médias ou ruins sugere que esses medicamentos podem possuir número reduzido de avaliações ou perfil de uso específico. Assim, o ranking deve ser interpretado como indicativo de alta aceitação, mas não como uma comparação definitiva entre todos os medicamentos disponíveis na base.

7.5 Pergunta 5 — Há concentração de avaliações positivas em poucos fabricantes?

Objetivo: avaliar se as avaliações excelentes estão concentradas em poucos fabricantes, calculando a participação dos Top 5 fabricantes (ordenados por média de avaliações excelentes) sobre o total de fabricantes.

37

```
# 7.5 - Concentração de avaliações positivas (Top N fabricantes) - Camada Gold

from pyspark.sql import functions as F

df_manu_gold = spark.table("workspace.default.medicine_manufacturer_gold")

# Top 5 fabricantes por avg_excellent_review
top_n = 5

df_top = (
    df_manu_gold
    .orderBy(F.col("avg_excellent_review").desc())
    .limit(top_n)
)

# Somatórios (métrica simples de concentração com base nas médias)
sum_top = df_top.agg(F.sum("avg_excellent_review").alias("sum_top")).collect()[0]["sum_top"]
sum_all = df_manu_gold.agg(F.sum("avg_excellent_review").alias("sum_all")).collect()[0]["sum_all"]

share_top = (sum_top / sum_all) * 100 if sum_all else None

display(df_top)

print(f"Participação aproximada do Top {top_n} (com base na soma das médias de Excellent_Review): {share_top:.2f}%")
```

df_manu_gold: pyspark.sql.connect.dataframe.DataFrame = [Manufacturer: string, avg_excellent_review: double ... 2 more fields]
df_top: pyspark.sql.connect.dataframe.DataFrame = [Manufacturer: string, avg_excellent_review: double ... 2 more fields]

Table

	Manufacturer	avg_excellent_review	avg_average_review	avg_poor_review
1	Clementia Pharmaceuticals Pvt L...	100	0	0
2	Gamete Pharma Pvt Ltd	100	0	0
3	Depsons Healthcare	100	0	0
4	Concord Drugs Ltd	100	0	0
5	Domagk Smith Labs Pvt Ltd	100	0	0

5 rows

```
Participação aproximada do Top 5 (com base na soma das médias de Excellent_Review): 1.59%
```

Comentário após o resultado (7.5):
A análise do Top 5 fabricantes indica que, embora esses fabricantes apresentem médias máximas de avaliações excelentes, a participação conjunta representa aproximadamente 1,59% do total considerado.

Esse resultado sugere **baixa concentração de avaliações positivas**, indicando que a percepção favorável dos usuários está distribuída entre diversos fabricantes, e não concentrada em poucos grupos dominantes. Ressalta-se que a métrica utilizada é aproximada, baseada na soma das médias, e análises futuras poderiam ponderar pelo volume de medicamentos ou avaliações por fabricante.

8. Conclusão e Considerações Finais

Este MVP teve como objetivo demonstrar a construção de um **pipeline de dados ponta a ponta**, utilizando a plataforma Databricks, para análise de avaliações de medicamentos a partir de dados públicos.

Ao longo do projeto, foram implementadas todas as etapas fundamentais de um fluxo de Engenharia de Dados:

- **Ingestão de dados (Camada Bronze):** carregamento dos dados brutos, preservando a estrutura original da fonte.
- **Transformação e padronização (Camada Silver):** limpeza de campos textuais, padronização de colunas e preparação dos dados para análise.
- **Análise de qualidade dos dados:** verificação de valores nulos, consistência dos percentuais de avaliação e identificação de limitações do conjunto de dados.
- **Modelagem analítica (Camada Gold):** criação de tabelas agregadas voltadas para responder perguntas de negócio.
- **Análises exploratórias:** resposta objetiva às perguntas de negócio propostas, com interpretação crítica dos resultados.

Os resultados obtidos indicam alta incidência de avaliações classificadas como excelentes, tanto para medicamentos quanto para fabricantes. Entretanto, a análise também evidenciou limitações importantes do dataset, como possíveis amostras reduzidas por grupo e baixa variabilidade em alguns atributos, o que restringe inferências estatísticas mais profundas.

Apesar dessas limitações, o MVP cumpre seu papel ao demonstrar domínio técnico na construção de pipelines de dados, aplicação de boas práticas de engenharia e capacidade de análise crítica sobre os resultados obtidos.

41

```python
# 8.2 - Visão geral do volume de dados analisados

from pyspark.sql import functions as F

df_bronze = spark.table("workspace.default.medicine_details")
df_silver = spark.table("workspace.default.medicine_details_silver")
df_gold = spark.table("workspace.default.medicine_manufacturer_gold")

summary_df = spark.createDataFrame(
    [
        ("Bronze", df_bronze.count()),
        ("Silver", df_silver.count()),
        ("Gold (fabricantes)", df_gold.count())
    ],
    ["Camada", "Quantidade de Registros"]
)

display(summary_df)
```

```
df_bronze: pyspark.sql.connect.dataframe.DataFrame = [Medicine Name: string, Composition: string ... 7 more fields]
df_gold: pyspark.sql.connect.dataframe.DataFrame = [Manufacturer: string, avg_excellent_review: double ... 2 more fields]
df_silver: pyspark.sql.connect.dataframe.DataFrame = [Medicine_Name: string, Composition: string ... 7 more fields]
summary_df: pyspark.sql.connect.dataframe.DataFrame = [Camada: string, Quantidade de Registros: long]
```

Table

	Camada	Quantidade de Registros
1	Bronze	11825
2	Silver	11825
3	Gold (fabricantes)	759

3 rows

Considerações Finais

A visão geral do volume de dados confirma a consistência do pipeline desenvolvido. Observa-se que a quantidade de registros foi preservada da Camada Bronze para a Camada Silver (11.825 registros), indicando que as etapas de limpeza e padronização não resultaram em perda de informações relevantes.

Na Camada Gold, a redução para 759 registros é esperada, uma vez que os dados foram agregados por fabricante, com o objetivo de viabilizar análises analíticas e responder às perguntas de negócio propostas.

Esse comportamento evidencia a correta aplicação do modelo de camadas (Bronze, Silver e Gold), reforçando a separação entre dados brutos, dados tratados e dados analíticos, conforme as boas práticas de Engenharia de Dados.

9. Autoavaliação e Possíveis Melhorias Futuras

A construção deste MVP permitiu aplicar, de forma prática, os conceitos de Engenharia de Dados abordados na disciplina, desde a ingestão até a análise analítica dos dados. Ao longo do desenvolvimento, foi possível identificar pontos fortes do projeto, bem como oportunidades claras de melhoria para trabalhos futuros.

Um dos principais aprendizados foi a importância da **qualidade e da diversidade dos dados**. Embora o conjunto utilizado tenha permitido a construção de um pipeline completo, observou-se baixa variabilidade em alguns atributos, especialmente nos percentuais de avaliações, o que limitou análises comparativas mais profundas. Em projetos futuros, a incorporação de bases com maior volume de avaliações por medicamento e fabricante poderia gerar resultados estatisticamente mais robustos.

Outro ponto de melhoria está relacionado à **modelagem dos dados**. Campos textuais como `Uses` e `Side_effects` apresentaram alta granularidade, o que dificultou análises categóricas mais estruturadas. Em trabalhos futuros, esses atributos poderiam ser normalizados por meio de técnicas de processamento de linguagem natural (NLP), categorização automática ou criação de tabelas dimensionais específicas.

Do ponto de vista analítico, o projeto poderia ser expandido com a aplicação de **métricas ponderadas**, considerando o número de avaliações por medicamento ou fabricante, reduzindo o impacto de amostras pequenas. Além disso, técnicas de visualização de dados e dashboards interativos poderiam ser incorporadas para facilitar a comunicação dos resultados a públicos não técnicos.

Por fim, este MVP cumpriu seu objetivo de demonstrar a construção de um pipeline de dados em nuvem, seguindo boas práticas de Engenharia de Dados. Para projetos futuros, pretende-se evoluir o nível de complexidade, incorporando novas fontes, validações mais avançadas de qualidade e métodos analíticos mais sofisticados, consolidando o aprendizado adquirido ao longo da disciplina.