📑 DOCUMENTO TÉCNICO — Consumer Profiling & Behavior Clustering

Versão: 1.0
Responsável: Ewerton Florencio
Tipo: Documento Técnico
Projeto: Segmentação de Consumidores por Comportamento
Escopo: *Data Science/Data Engineering*

1. Arquitetura Técnica

A solução será organizada em três camadas:

- **Raw:** Armazenamento da fonte original sem alterações.
- **Silver:** Aplicação de regras de limpeza, padronização e ajustes básicos.
- **Gold:** Agregações analíticas, construção das métricas **RFM** e preparação para modelagem.

O processamento poderá ocorrer em ambiente **local** ou **Databricks Community**, utilizando **Python** e **PySpark**.

2. Fontes de Dados

Detalhe	**Especificação**
Fonte principal	Online Retail Dataset (Kaggle)
Formato	CSV
Campos originais	InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country
Observação	Nenhum dado sensível é utilizado.

3. Regras de Transformação — Silver

3.1 Conversão de Tipos

Campo	Tipo Alvo
InvoiceDate	datetime
Quantity	integer
UnitPrice	float
CustomerID	string

3.2 Limpeza e Padronização

- Remover registros sem **CustomerID**.
- Remover linhas onde **Quantity <= 0**, exceto se representarem devolução claramente identificável.
- Remover registros com **UnitPrice <= 0**.
- Remover duplicidades por combinação de: **InvoiceNo, StockCode, InvoiceDate, CustomerID**.

3.3 Enriquecimentos

- Criação da coluna **TotalPrice** = **Quantity** * **UnitPrice**.
- Padronização de textos (opcional).

4. Regras da Camada Gold

A camada Gold consolida as métricas por cliente.

4.1 Recency

- **Identificar** a data de referência (última data presente no dataset).
- **Recency** = diferença em dias entre a referência e a última compra do cliente.

4.2 Frequency

- **Frequency** = quantidade de faturas distintas (**InvoiceNo**) associadas ao cliente.

4.3 Monetary

- **Monetary** = soma de todos os **TotalPrice** do cliente.

4.4 Normalização

As métricas **Recency, Frequency e Monetary** serão normalizadas utilizando **padrão z-score** ou **StandardScaler**.

5. Modelagem Analítica

5.1 Algoritmo principal

- **K-Means**.
- **Justificativa técnica:** funciona de forma adequada em dados normalizados e para segmentações comportamentais.

5.2 Alternativo

- **DBSCAN** (para análise complementar de densidade, caso necessário).

5.3 Critérios de definição de K

- Avaliação pelo método **Elbow**.
- Validação pelo **Silhouette Score**.

5.4 Parâmetros de treinamento (padrão)

Parâmetro	Valor
max_iter	300
n_init	10
random_state	42

5.5 Saída da modelagem

- Atribuição de **cluster** por cliente.
- Métricas agregadas por cluster (médias e distribuições).

6. Outputs Técnicos

6.1 Conjuntos de dados finais

- Tabela contendo métricas **RFM normalizadas**.
- Tabela contendo a **classificação final** dos clientes em clusters.

6.2 Artefatos técnicos

- Notebooks de ingestão, transformação, agregação, modelagem e visualização.
- Código modular em pasta src/ para reutilização.
- Arquivo requirements.txt com dependências.

7. Estrutura do Repositório (GitHub)

```
/
├── data/
│   ├── raw/
│   ├── silver/
│   └── gold/
├── notebooks/
│   ├── 01_ingestao_raw.ipynb
│   ├── 02_silver_transform.ipynb
│   ├── 03_gold_rfm.ipynb
│   ├── 04_modelo_cluster.ipynb
│   └── 05_visualizacoes.ipynb
├── src/
│   ├── preprocess.py
│   ├── rfm.py
│   ├── clustering.py
│   └── plots.py
├── docs/
├── README.md
└── requirements.txt
```

8. Tecnologias

- **Python**
- **PySpark** (opcional, dependendo do volume)
- **Pandas**
- **Scikit-learn**
- **Matplotlib / Seaborn**
- **Git / GitHub**

9. Plano de Testes Técnicos

9.1 Testes de qualidade dos dados

- Validação de tipos.
- Verificação de duplicidades.
- Consistência de datas.
- Valores negativos ou impossíveis.

9.2 Testes da camada Gold

- **Recency** calculado corretamente.
- **Frequency** refletindo o número de faturas distintas.
- **Monetary** correspondente ao somatório real.

9.3 Testes do modelo

- Execução do **K-Means** sem falhas.
- **Silhouette Score** acima do mínimo aceitável.
- Distribuição dos clusters coerente.

10. Métricas Técnicas de Avaliação

- Inertia (Elbow)
- Silhouette Score
- Distribuição percentual dos clusters
- Variância intra-cluster

11. Critérios Técnicos de Aceite

O projeto será considerado tecnicamente aceito quando:

1. O **pipeline executa integralmente** (Raw → Silver → Gold → Modelo).
2. As métricas **RFM são reproducíveis**.
3. O modelo de clusterização gera **grupos interpretáveis**.
4. Os **notebooks executam sem erros**.
5. A **organização do repositório** segue o padrão definido.

12. Itens Fora do Escopo Técnico

- Integração com sistemas externos.
- Deploy do modelo em API.
- Dashboards Power BI.
- Processamento distribuído em cluster real.
- Orquestração via Airflow.