UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

Relatório da Disciplina MT803

Segmentação de Clientes de Cafeteria via K-Prototypes: Um Estudo de Caso com Dados "Dirty Cafe Sales"

Aluno: Matheus Queiroz Mota, **RA:** 251495

1 Introdução

O presente trabalho teve como base o dataset "Dirty Cafe Sales" disponível no repositório Kaggle[1]. Trata-se de cerca de 10 000 registros de transações de uma cafeteria fictícia, intencionalmente "*dirty*": valores ausentes, tokens "ERROR"/"UNKNOWN", preços zerados e datas inválidas, buscando assim simular problemas reais de qualidade de dados em pontos de venda.

Para lidar com essas inconsistências, desenvolvemos uma *pipeline* de pré-processamento composta por:

- Padronização de tokens irregulares para `NaN`;
- Conversão de colunas numéricas e de data para tipos adequados;
- Recalculo de *Total Spent* em casos de incoerência;
- Criação de atributos derivados (*ticket médio por item*, *trimestre da venda*);
- Imputação de valores faltantes (mediana para numéricos; categoria "Unknown" para categóricos) e remoção de registros sem *Item*.

Com os dados devidamente limpos, aplicou-se o algoritmo *K-Prototypes* — que lida simultaneamente com variáveis numéricas e categóricas — para segmentar clientes em perfis de consumo. Testamos k de 2 a 8 e selecionamos $k = 2$ com base no índice de Silhouette ($\approx 0{,}554$). Os dois clusters revelaram perfis consistentes (clientes "pega-e-leva" versus consumidores de "refeições completas"), oferecendo subsídios para ações de marketing, fidelização e projeções de revenue.

[1] `https://www.kaggle.com/datasets/ahmedmohamed2003/cafe-sales-dirty-data-for-cleaning-training`

Este relatório detalha a metodologia de limpeza, a modelagem de clusterização, os principais resultados e as recomendações práticas para a PC315, além de discutir limitações dos dados.

2 Descrição do Banco de Dados

O dataset "Dirty Cafe Sales" contém 10000 registros de vendas, com variáveis brutas e derivadas:

- **Transaction ID** (string): identificador único da transação.
- **Item** (categórica): nome do produto (e.g., Coffee, Sandwich).
- **Quantity** (inteira): quantidade de unidades adquiridas.
- **Price Per Unit** (numérica): preço unitário do item (U$).
- **Total Spent** (numérica): gasto total na transação.
- **Payment Method** (categórica): forma de pagamento (Cash, Credit Card, etc.).
- **Location** (categórica): local de consumo (In-store, Takeaway).
- **Transaction Date** (data): data da compra (YYYY-MM-DD).
- **Quarter** (inteiro): trimestre da venda (derivada de `Transaction Date`).

Variável	**Irregulares (%)**
Location	39,6 %
Payment Method	31,8 %
Item	9,7 %
Price Per Unit	5,3 %
Total Spent	5,0 %
Quantity	4,8 %
Transaction Date	4,6 %
Quarter	4,6 %
Transaction ID	0,0 %

Tabela 1: Percentual de registros irregulares por variável (dados brutos).

Problema	Ocorrências	Ação corretiva
Tokens inválidos (‘ERROR‘, ‘UNKNOWN‘, . . .)	10082	Substituir por `NaN`
Falha na conversão numérica	1514	Imputar mediana
`Total Spent` incoerente	0	Recalcular Quantity × Price Per Unit
Datas inválidas (NaT)	460	Remover linhas com data inválida
‘Item‘ ausente	969	Descartar linhas sem ‘Item‘
Location faltante	3961	Preencher com “Unknown”
Payment Method faltante	3178	Preencher com “Unknown”

Tabela 2: Inconsistências detectadas no dataset bruto e respectivas ações corretivas.

3 Pipeline de Tratamento e Execução dos Dados

Passo 1: Importar CSV Bruto
Carregar o arquivo `dirty_cafe_sales.csv`. *Se falhar, interromper o processo.*

Passo 2: Converter Tipos

- *Quantity*, *Price Per Unit*, *Total Spent* → `float`
- *Transaction Date* → `datetime` via `pd.to_datetime(..., errors='coerce')`

Passo 3: Padronizar Dados Irregulares

- *Numéricas*: detectar `NaN`, imputar pela mediana.
- *Textuais*: normalizar para minúsculas; tokens `ERROR`, `UNKNOWN`, `None` e strings vazias → “Unknown”.
- *Datas*: entradas inválidas transformadas em `NaT` (serão removidas no Passo 4).

Passo 4: Remoção de Dados Inválidos

- Descatar linhas com *Item* = “Unknown”.
- Descatar linhas com *Transaction Date* = `NaT`.

Passo 5: Verificar e Reconciliar *Total Spent*
Se *Total Spent* estiver vazio ou diferente de *Quantity* × *Price Per Unit*, *Total Spent* recebe *Quantity* × *Price Per Unit*. *Em caso de NaN residual, retorne ao Passo 3.*

Passo 6: Criar Atributos Derivados

– *Ticket Médio por Item = Total Spent / Quantity.*

– *Quarter* extraído de *Transaction Date*.

Passo 7: Clusterização

– Aplicar **K-Prototypes** (dados numéricos + categóricos).

– Avaliar $k = 2 \ldots 8$ pelo Índice de Silhouette; escolher $k = 2$ (Silhouette = 0,554).

Passo 8: Exportar Dataset Limpo & Clusters
Salvar em `dirty_cafe_sales_cleaned_clusters.csv`, contendo apenas as variáveis finais: `Transaction ID`, `Item`, `Quantity`, `Price Per Unit`, `Total Spent`, `Payment Method`, `Location`, `Transaction Date`, `Quarter`, `AvgSpendingPerItem`, `Cluster` e `Semester`. *Esta etapa consolida todo o pré-processamento e atribui a cada registro o rótulo de segmento obtido pela clusterização.*

4 Análise Exploratória — Distribuição e Correlações

Antes de aplicar o algoritmo de clusterização, examinamos a distribuição de gasto total por transação e as correlações entre nossas variáveis originais.

4.1 Distribuição de *Total Spent*

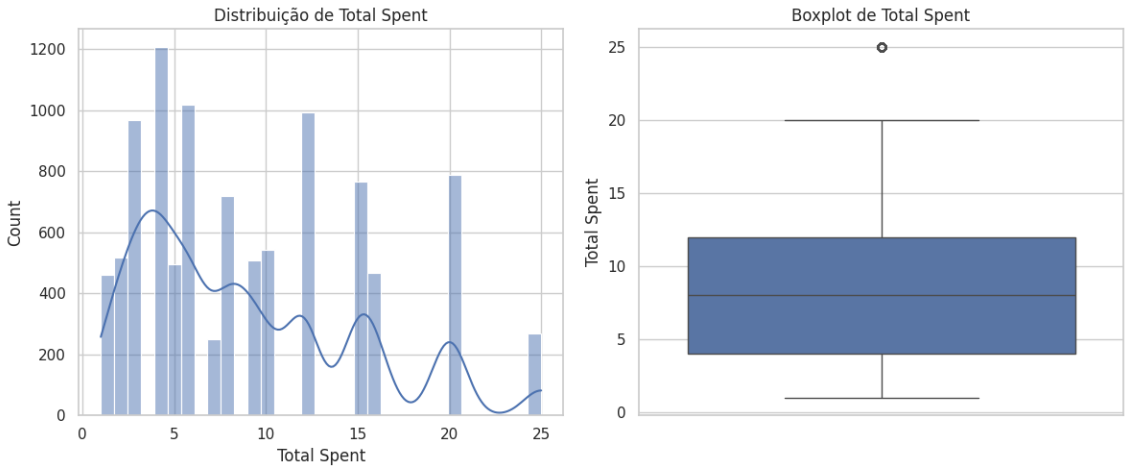
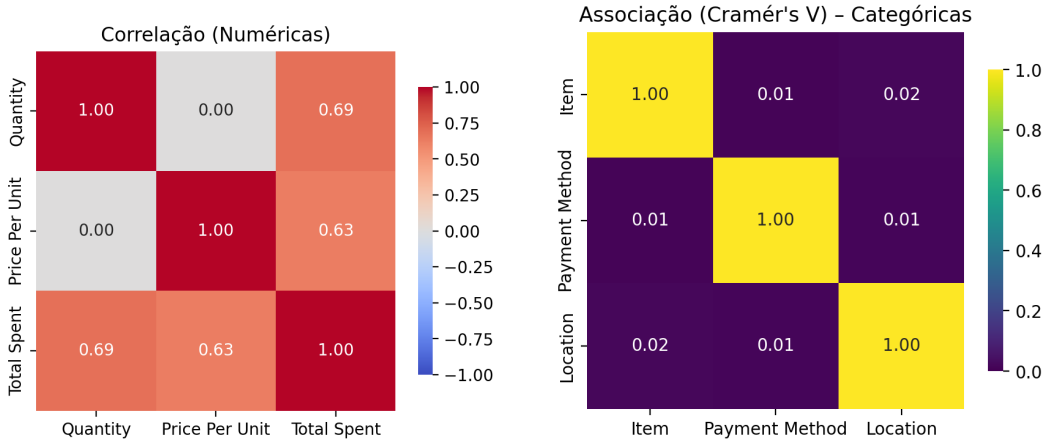


Figura 1: Histograma e boxplot de *Total Spent* em U$ por transação, após limpeza. Note que o valor varia de U$1 a U$25, com mediana em U$8 e alguns outliers acima de U$20.

As variáveis *Price Per Unit* e *Quantity* também foram analisadas, mas apresentam distribuição semelhante em U$1–U$5 e 1–5 itens respectivamente, sem outliers extremos que justifiquem visualização adicional.

4.2 Correlação e Associação



(a) Correlação de Pearson (numéricas) (b) Cramér's V (categóricas)

Figura 2: (a) Correlação entre *Quantity*, *Price Per Unit* e *Total Spent* (–1 a 1). (b) Associação entre *Item*, *Payment Method* e *Location* (0 a 1).

Principais insights:

- corr(Quantity, Total Spent) = 0,69: aumentar número de itens tende a elevar o gasto total.
- corr(Price Per Unit, Total Spent) = 0,63: itens mais caros geram maior receita.
- corr(Quantity, Price Per Unit) ≈ 0: preço unitário não influencia quantidade vendida.
- Máximo Cramér's V ≈ 0,03: *Item*, *Payment Method* e *Location* praticamente independentes, indicando sinais distintos para o modelo de clusterização.

5 Aprendizado de Máquina — Segmentação de Clientes

Para identificar perfis de consumo distintos, aplicamos um algoritmo de *clustering* capaz de lidar simultaneamente com variáveis numéricas e categóricas.

5.1 Features Utilizadas

- **Numéricas:**
  - *Quantity* — número de itens por transação.
  - *Price Per Unit* — preço unitário em U$.
  - *Total Spent* — gasto total por transação em U$.

  - *AvgSpendingPerItem* — gasto médio por item (derivada).
  - *Quarter* — trimestre da venda (1–4).

- **Categóricas:**
  - *Item* — tipo de produto (ex.: "Coffee", "Sandwich").
  - *Payment Method* — forma de pagamento ("Cash", "Credit Card", "Digital Wallet", "Unknown").
  - *Location* — local de consumo ("In-store", "Takeaway", "Unknown").

5.2 Transformações Aplicadas

1. **Imputação:** *Numéricas* com mediana; *categóricas* (tokens irregulares como `ERROR`, `None`, strings vazias) agrupadas em "Unknown".
2. **Derivação de atributos:** cálculo de *AvgSpendingPerItem* = Total Spent / Quantity e extração de *Quarter* da data.
3. **Conversão:** variáveis numéricas mantidas na escala original; variáveis categóricas processadas internamente pelo K-Prototypes.

5.3 Escolha do Algoritmo

Optamos pelo *K-Prototypes* (biblioteca `kmodes`) pois:

- **Dados mistos:** combina distância Euclidiana (numéricos) e dissimilaridade χ^2 (categóricos).
- **Eficiência:** dispensa codificação one-hot e preserva similaridades de cada tipo de variável.
- **Validação:** testamos $k = 2 \dots 8$ via coeficiente de *silhouette*, escolhendo $k = 2$ com valor máximo 0,554.

Esse modelo supera K-Means (só numérico) e K-Modes (só categórico), segmentando com fidelidade o comportamento de compra da PC315.

6 Resultados dos Clusters

Para $k = 2$, escolhemos o valor com melhor *silhouette*:

k	2	3	4	5	6	7	8
Silhouette	**0,554**	0,444	0,403	0,391	0,373	0,348	0,377

Tabela 3: Coeficiente de *silhouette* para diferentes valores de k.

6.1 KPIs por Cluster

Cluster	% Vendas	Ticket Médio (U$)	% Receita
0	63 %	5,0	40 %
1	37 %	16,0	60 %

Tabela 4: Principais métricas por cluster.

6.2 Perfis de Consumo

- **Cluster 0 – Pega-e-Leva:** 63 % das vendas, ticket baixo (≈U$5), foco em bebidas e snacks "to-go", alta taxa de "Unknown". *Recomendação:* combos rápido (bebida+snack) e incentivo ao cadastro.
- **Cluster 1 – Refeição Completa:** 37 % das vendas, 60 % da receita, ticket ≈U$16, refeições e sobremesas no salão, uso frequente de cartão. *Recomendação:* combos premium (refeição+bebida), programa de pontos.

7 Validação Estatística

Comparamos "Unknown" vs "Known" em *Location* e *Payment Method* para *Quantity*, *Price Per Unit* e *Total Spent*:

1. Formular $H_0 : \mu_{\text{Unknown}} = \mu_{\text{Known}}$ vs $H_1 : \mu_{\text{Unknown}} \neq \mu_{\text{Known}}$.
2. Teste de normalidade (Shapiro–Wilk): se $p > 0{,}05$, *t-test*; caso contrário, Mann–Whitney.
3. Calcular p-valor para cada métrica e variável categórica.
4. Interpretar: $p < 0{,}05 \Rightarrow$ diferença significativa; $p \geq 0{,}05 \Rightarrow$ não significativa.
5. Manter "Unknown" mesmo sem diferença ($p \geq 0{,}05$), pois sinaliza falha de captura e evita viés de imputação.

8 Conclusão e Recomendações

Este estudo demonstrou que, mesmo em meio a registros incompletos, é possível extrair perfis de consumo claros e acionáveis. A seguir, segue um resumo os principais pontos e sugerimos próximos passos:

- **Qualidade dos Dados:**
  - Alta proporção de *Unknown* em *Location* (40 %) e *Payment Method* (32 %) — sinaliza falhas no PDV/app (Ponto de Venda / aplicativo).
  - Futuras coletas devem incluir *Customer ID* e carimbo de data-hora completo para análises de recorrência e sazonalidade.

- **Modelagem de Clusters:**
  - $k = 2$ apresentou boa separação (*silhouette* = 0,555), porém explorar $k > 2$, *DBSCAN* ou *GMM* pode revelar subgrupos de valor.
  - Analisar outras métricas (por ex. *Calinski–Harabasz*, *Davies–Bouldin*) para complementar a escolha de k.
- **Validação Empírica:**
  - Implementar testes A/B de promoções por cluster e calcular *lifetime value* para aferir impacto real.
  - Monitorar KPIs antes e depois das ações para ajustar estratégias.
- **Aperfeiçoamento de Features e EDA:**
  - Criar variáveis de frequência de compra, ticket recorrente e sazonalidade (mês, hora, dia da semana).
  - Integrar informações de mix de produtos e locais de consumo, para assimilar mais o perfil dos clientes

Em síntese, a aplicação de *K-Prototypes* forneceu dois perfis estratégicos — *Pega-e-Leva* e *Refeição Completa* — que podem orientar promoções e programas de fidelização. Ao abordar as limitações de dados e aprofundar a validação, a PC315 estará melhor equipada para maximizar receita e engajamento de seus clientes.

9 Referências e Recursos

- **Bibliotecas Python (versões):**
  - `streamlit` — 1.34.0
  - `streamlit-option-menu` — 0.3.5
  - `pandas` — 2.2.2
  - `numpy` — 1.26.4
  - `scipy` — 1.11.4
  - `plotly` — 5.19.0
  - `matplotlib` — 3.8.4
  - `seaborn` — 0.13.2
- **Código-fonte e documentação:**
  `https://github.com/Matiass51752/251495_MT803_Trabalho_Final`
- **Dashboard interativo:**
  `https://251495mt803trabalhofinal-dsnjjwc3xqighspddd2d8g.streamlit.app/`