

Curso  
**ANÁLISE E  
DESENVOLVIMENTO DE  
SISTEMAS**

Disciplina  
**INTELIGÊNCIA ARTIFICIAL E MACHINE LEARNING**

Professor  
**LUCY MARI**

Data  
**10/11/2025**

## **PROJETO INTERDISCIPLINAR – CANNOLI**

### **Integrantes:**

Anderson Yavi Fernandez - RA: 24025678

Gabriel Gonçalves Pires - RA: 24026518

Isabela Nunes Zeferino - RA: 24026460

Kaique Neres de Oliveira - RA: 24026134

Luiz Felipe Galdino de Carvalho - RA: 2402656

## 1. Introdução e Objetivo

O principal desafio de marketing da Cannoli é a falta de personalização, pois a base de clientes não é segmentada. Isso leva a campanhas de marketing genéricas e pouco eficientes. O objetivo deste trabalho foi aplicar um modelo de **Machine Learning (ML) K-Means** para resolver esse problema, agrupando os clientes em perfis distintos com base no seu comportamento de compra (Análise RFM), permitindo a criação de estratégias de marketing mais inteligentes e direcionadas.

## 2. Metodologia: Preparação e Aplicação do Modelo

### 2.1 Preparação e Carregamento da Base ETL

Para garantir a acurácia e relevância da análise, o modelo ML foi executado sobre o arquivo **mock\_orders\_for\_rfm.csv**, que representa a base de **5.000 pedidos limpos e tratados** pelo processo de ETL do grupo. Esta base contém apenas os parâmetros necessários para o cálculo do RFM (Recência, Frequência e Valor Monetário).

**Código de Carregamento, Cálculo RFM e Normalização:**

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from datetime import datetime, timedelta

# 1. Carregando a base de pedidos LIMPA gerada pelo ETL
df_order = pd.read_csv('mock_orders_for_rfm.csv', sep=';', decimal=',')
df_order['createdAt'] = pd.to_datetime(df_order['createdAt'],
                                         dayfirst=True, errors='coerce')

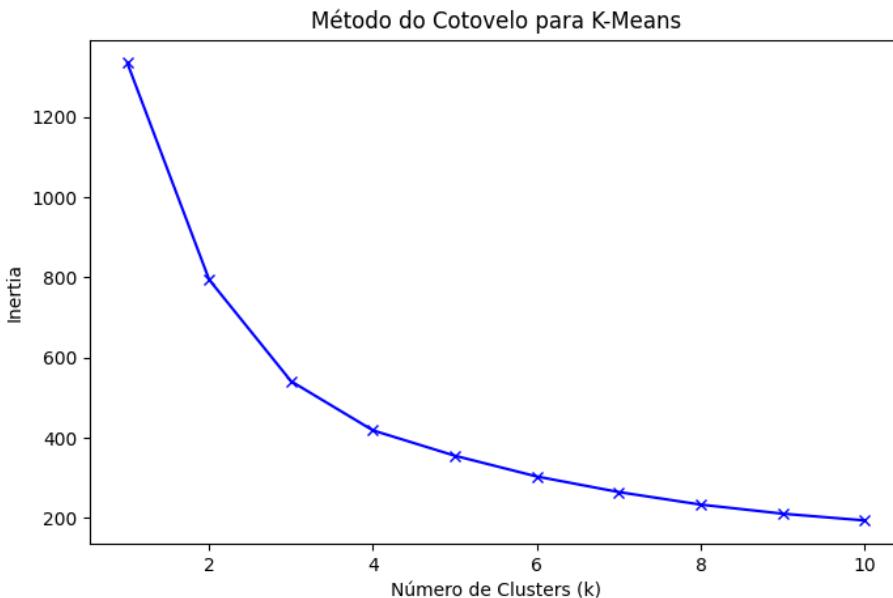
# 2. Cálculo das Métricas RFM (Recência, Frequência, Monetário)
data_maxima = df_order['createdAt'].max() + timedelta(days=1)
rfm_df = df_order.groupby('customer').agg(
    Recencia=('createdAt', lambda date: (data_maxima -
                                         date.max()).days),
    Frequencia=('id', 'count'),
    Monetario=('totalAmount', 'sum'))
rfm_df.reset_index()

# 3. Normalização dos Dados
rfm_features = rfm_df[['Recencia', 'Frequencia', 'Monetario']]
scaler = StandardScaler()
rfm_scaled = scaler.fit_transform(rfm_features)
```

## 2.2 Descoberta e Aplicação do K-Means

Para determinar o número ideal de grupos, foi utilizado o **Método do Cotovelo** (Elbow Method), que mede a inércia (soma das distâncias quadradas dentro do cluster). O ponto onde a queda da inércia se torna menos acentuada (o "cotovelo") indica o valor ótimo de k. Neste caso, o valor ideal foi determinado como **k=4**.

**Gráfico: Método do Cotovelo (Determinação do k)**



O algoritmo K-Means foi aplicado com  $k=4$ , e o resultado foi ranqueado e nomeado para facilitar a interpretação de negócios.

### Código: Aplicação K-Means e Ranqueamento dos Segmentos

```
# Aplicação do Modelo Final (k=4)
kmeans_final = KMeans(n_clusters=4, random_state=42, n_init=10)
kmeans_final.fit(rfm_scaled)
rfm_df['Cluster'] = kmeans_final.labels_

# Análise e Ranqueamento dos Segmentos (Conclusão de Negócio)
analise_clusters = rfm_df.groupby('Cluster')[['Recencia',
'Frequencia', 'Monetario']].mean().reset_index()

analise_clusters['rank_recencia'] =
analise_clusters['Recencia'].rank(ascending=True)
analise_clusters['rank_frequencia'] =
analise_clusters['Frequencia'].rank(ascending=False)
analise_clusters['rank_monetario'] =
analise_clusters['Monetario'].rank(ascending=False)
analise_clusters['pontuacao_geral'] =
analise_clusters['rank_recencia'] +
analise_clusters['rank_frequencia'] +
analise_clusters['rank_monetario']
analise_clusters = analise_clusters.sort_values('pontuacao_geral')
nomes_segmentos = ['Clientes Campeões', 'Clientes Leais', 'Clientes Novos/Promissores', 'Clientes em Risco']
analise_clusters['Segmento'] =
nomes_segmentos[:len(analise_clusters)]
```

### 3. Resultados e Conclusão

A aplicação do modelo K-Means foi bem-sucedida em segmentar a base de clientes em quatro grupos acionáveis, conforme detalhado na tabela de resultados a seguir. O ranqueamento final, feito na etapa de código, permite identificar de forma clara o valor estratégico de cada grupo.

**Tabela de Resumo dos Segmentos:**

--- RESULTADO FINAL: Resumo dos Segmentos Encontrados ---				
	Segmento	Recencia	Frequencia	Monetario
3	Clientes Campeões	36.056604	17.754717	3397.449434
0	Clientes Leais	35.145570	12.626582	2370.631582
1	Clientes Novos/Promissores	173.173333	9.480000	1779.974267
2	Clientes em Risco	45.603774	8.509434	1485.816478

O ranqueamento e a nomeação dos segmentos foram essenciais para transformar o resultado matemático do K-Means em inteligência de negócio, entregando a solução para o desafio de marketing da Cannoli. O projeto demonstra a capacidade de utilizar o Machine Learning sobre uma base de dados segmentada e tratada (ETL) para desenvolver estratégias de marketing mais direcionadas e eficientes.