**📈 Página 5: Detecção de Padrões e Anomalias**

**🔎 Visão Geral da Detecção de Anomalias**

Nesta etapa, aplicamos técnicas avançadas para identificar padrões incomuns e anomalias que podem indicar possíveis fraudes ou falhas sistêmicas nas entregas do Walmart. Usamos técnicas modernas como análise estatística, clusterização e identificação de outliers.

Utilizamos Python com bibliotecas como Scikit-learn, Seaborn e Matplotlib para conduzir essa análise detalhada.

**📉 Técnicas Utilizadas**

**1. Clusterização com K-Means**

Utilizamos o algoritmo K-Means para segmentar motoristas com comportamento semelhante e identificar grupos com comportamento suspeito:

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# Selecionando variáveis relevantes

drivers\_behavior = orders\_df.groupby('driver\_id').agg({

'items\_missing': 'mean',

'order\_amount': 'mean',

'items\_delivered': 'mean'

}).reset\_index()

# Normalizando os dados

scaler = StandardScaler()

drivers\_scaled = scaler.fit\_transform(drivers\_behavior[['items\_missing', 'order\_amount', 'items\_delivered']])

# Aplicando K-Means

kmeans = KMeans(n\_clusters=3, random\_state=42)

drivers\_behavior['cluster'] = kmeans.fit\_predict(drivers\_scaled)

# Visualizando os clusters

plt.figure(figsize=(10,6))

sns.scatterplot(data=drivers\_behavior, x='order\_amount', y='items\_missing', hue='cluster', palette='Set1')

plt.title('Clusterização dos Motoristas')

plt.xlabel('Valor Médio dos Pedidos')

plt.ylabel('Média de Itens Faltantes')

plt.show()

**2. Análise de Outliers com Boxplot**

Boxplots são utilizados para detectar valores extremos ou suspeitos nas variáveis-chave:

plt.figure(figsize=(10,6))

sns.boxplot(data=orders\_df, x='items\_missing')

plt.title('Identificação de Outliers nos Itens Faltantes')

plt.xlabel('Número de Itens Faltantes')

plt.show()

**🕵️‍♂️ Padrões de Fraude Identificados**

**3. Motoristas com Alta Frequência de Reclamações**

Destacamos motoristas com comportamento suspeito:

# Motoristas com maior frequência relativa de itens faltantes

suspicious\_drivers = drivers\_behavior[drivers\_behavior['items\_missing'] > drivers\_behavior['items\_missing'].mean() + 2 \* drivers\_behavior['items\_missing'].std()]

print(suspicious\_drivers)

**4. Clientes com Padrões Recorrentes**

Analisamos clientes com frequência recorrente de reclamações:

# Clientes com alta recorrência de reclamações

suspicious\_customers = orders\_df.groupby('customer\_id')['items\_missing'].mean().sort\_values(ascending=False).head(10)

plt.figure(figsize=(10,6))

sns.barplot(x=suspicious\_customers.values, y=suspicious\_customers.index)

plt.title('Clientes com Maior Média de Itens Faltantes por Pedido')

plt.xlabel('Média de Itens Faltantes')

plt.ylabel('ID do Cliente')

plt.show()

**5. Horários Críticos**

Identificamos horários específicos que apresentam maior incidência de problemas:

# Taxa de itens faltantes por hora

hourly\_missing\_rate = orders\_df.groupby('delivery\_hour')['items\_missing'].mean()

plt.figure(figsize=(10,6))

sns.lineplot(x=hourly\_missing\_rate.index, y=hourly\_missing\_rate.values, marker='o')

plt.title('Taxa Média de Itens Faltantes por Hora do Dia')

plt.xlabel('Hora do Dia')

plt.ylabel('Média de Itens Faltantes')

plt.grid(True)

plt.show()

**📌 Principais Insights**

* Identificação clara de motoristas e clientes com comportamento suspeito.
* Destaque para horários críticos com taxas elevadas de anomalias.
* Evidências que sugerem possíveis fraudes ou falhas no processo de entrega.