Análise de Transações de Vendas para Detecção de Fraudes

Lucas Carvalho da Luz Moura

2025-09-10

Contents

1 1. Introdução

Este projeto tem como objetivo principal analisar um grande conjunto de dados de transações de vendas para identificar padrões e, especialmente, detectar transações potenciais fraudulentas. Para isso, utilizaremos uma abordagem multimétodo, que inclui técnicas clássicas de detecção de outliers, agrupamento (clustering) para segmentação de dados e modelagem preditiva semissupervisionada para estimar o risco de fraude em registros não inspecionados. Finalmente, faremos uma comparação com abordagens de Inteligência Artificial generativa, apontando as vantagens e limitações de cada método.

2 2. Carregamento e Preparação dos Dados

Nesta etapa, iniciamos o processo carregando o conjunto de dados limpos, obtidos após tratamento prévio para lidar com valores ausentes e inconsistências. Exploraremos rapidamente a estrutura dos dados, incluindo dimensões e exemplos iniciais, para entender as variáveis disponíveis e o volume dos registros que serão analisados.

```
# Carregar dados
load("salesClean.Rdata")

# Estrutura inicial
cat("Dimensões:", dim(sales), "\n")
```

```
## Dimensões: 400204 6
```

```
head(sales)
```

```
##   ID Prod Quant   Val Insp    Uprice
## 1 v1   p1   182  1665 unkn  9.148352
## 2 v2   p1  3072  8780 unkn  2.858073
## 3 v3   p1 20393 76990 unkn  3.775315
## 4 v4   p1   112  1100 unkn  9.821429
## 5 v3   p1  6164 20260 unkn  3.286827
## 6 v5   p2   104  1155 unkn 11.105769
```

3 3. Análise Exploratória

Com os dados carregados, realizamos uma análise exploratória sumária para detectar características gerais, distribuição dos dados, e a situação das variáveis-chave, como o status da inspeção (Insp). Essa etapa é fundamental para identificar possíveis desequilíbrios no dataset ou anomalias iniciais que impactam as análises subsequentes.

```
summary(sales)
```

```
##        ID              Prod            Quant                Val
##  v431   :  9811   p1125  :  3912   Min.   :        1   Min.   :     56.3
##  v54    :  6017   p3774  :  1823   1st Qu.:      106   1st Qu.:   1340.0
##  v426   :  3881   p1437  :  1703   Median :      160   Median :   2670.0
##  v1679  :  3016   p1917  :  1691   Mean   :     8172   Mean   :  14606.0
##  v1085  :  2985   p4089  :  1594   3rd Qu.:      683   3rd Qu.:   8675.0
##  v1183  :  2641   p2742  :  1517   Max.   :473883883   Max.   :4642955.0
##  (Other):371853   (Other):387964
##     Insp            Uprice
##  ok   : 14458   Min.   :    0.000
##  unkn :384478   1st Qu.:    8.712
##  fraud:  1268   Median :   12.027
##                 Mean   :   20.319
##                 3rd Qu.:   19.018
##                 Max.   :26460.700
##
```

```r
table(sales$Insp)
```

```
##
##     ok   unkn  fraud
##  14458 384478   1268
```

4 4. Detecção de Outliers

Outliers são observações que se distanciam significativamente do padrão geral dos dados, podendo indicar erros, situações atípicas ou mesmo fraudes. Para identificá-los, utilizamos uma técnica robusta baseada na diferença entre os quartis 1 e 3, conhecida como intervalo interquartílico (IQR). Essa técnica considera como outliers os valores que estejam muito abaixo do valor do primeiro quartil, subtraído de 1,5 vezes o IQR, ou muito acima do terceiro quartil, somado de 1,5 vezes o IQR. Essa regra é amplamente utilizada pela sua eficácia em detectar anomalias sem ser excessivamente sensível a dados extremos.

```r
detect_outliers <- function(x) {
  q <- quantile(x, probs = c(0.25, 0.75), na.rm = TRUE)
  iqr <- q[2] - q[1]
  lower <- q[1] - 1.5 * iqr
  upper <- q[2] + 1.5 * iqr
  x < lower | x > upper
}

outliers <- sum(detect_outliers(sales$Uprice), na.rm = TRUE)
cat("Total de outliers detectados:", outliers, "\n")
```

```
## Total de outliers detectados: 38578
```

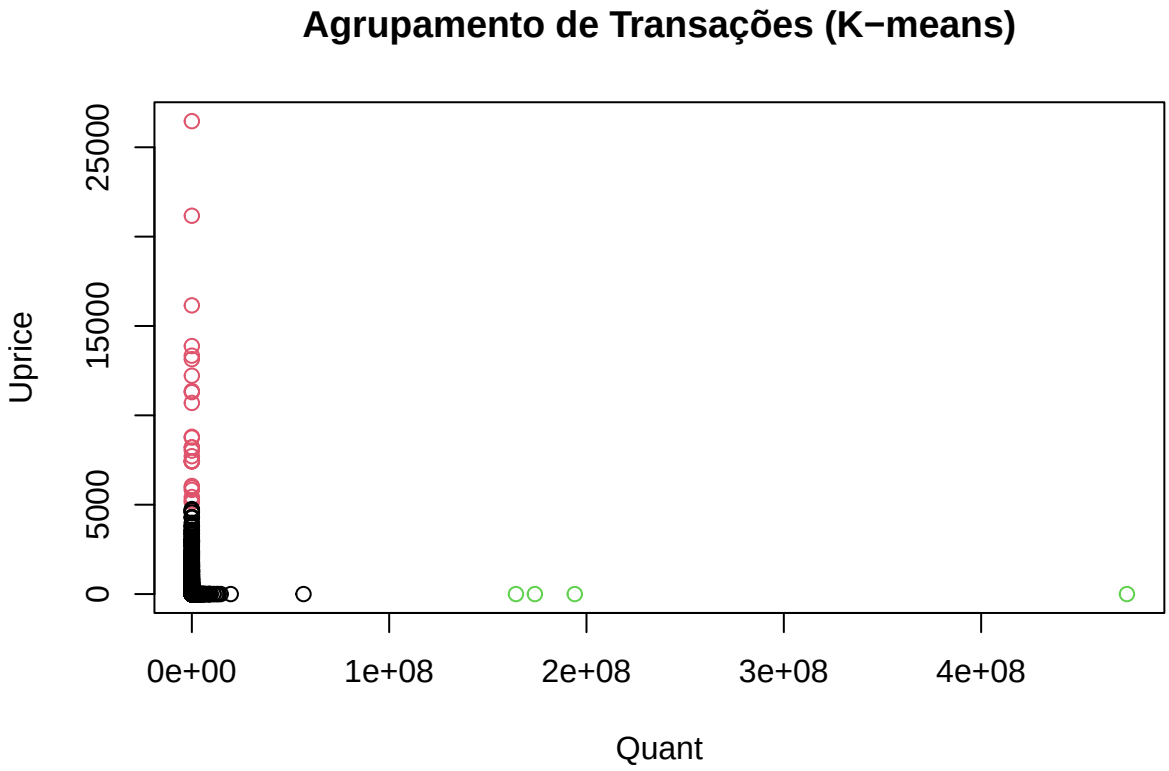
5 5. Clustering (Agrupamento)

O agrupamento via K-means tem como objetivo identificar padrões e segmentar o conjunto de transações em grupos homogêneos com base nas variáveis quantitativas 'Quantidade' e 'Preço Unitário'. Esse método é não supervisionado e auxilia na visualização e compreensão de grupos naturais nos dados, podendo destacar segmentos que contenham comportamentos suspeitos ou incomuns.

A aplicação do K-means consiste em padronizar os dados para evitar viés pela escala das variáveis e definir o número de clusters, aqui fixado em 3, para balancear simplicidade e discriminação entre grupos.

```r
set.seed(123)

# Guardar índices
valid_idx <- complete.cases(sales[, c("Quant", "Uprice")])

# Subconjunto sem NA
cluster_data <- sales[valid_idx, c("Quant", "Uprice")]
cluster_scaled <- scale(cluster_data)

# K-means
set.seed(123)
```

```r
kmeans_res <- kmeans(cluster_scaled, centers = 3, nstart = 25)

# Colocar resultados de volta no sales
sales$Cluster <- NA
sales$Cluster[valid_idx] <- kmeans_res$cluster

# Visualização
plot(cluster_data, col = kmeans_res$cluster,
     main = "Agrupamento de Transações (K-means)")
```



6 6. Sistema de Detecção de Fraude Baseado em Regras

Nesta fase, construímos uma regra baseada no cálculo do escore Z para o preço unitário por produto, que compara o preço de cada transação com a média e o desvio padrão do produto correspondente. Transações com escore Z absoluto superior a 3 são consideradas de alto risco, pois indicam valores atípicos estatisticamente significativos.

Este sistema simples porém eficaz permite classificar automaticamente as transações conforme seu risco, auxiliando na priorização para inspeção manual ou análise automatizada.

```r
# Calcular escore baseado em Z-score
prod_stats <- aggregate(Uprice ~ Prod, data = sales,
                        function(x) c(mean = mean(x), sd = sd(x)))
```

```r
sales_stats <- merge(sales, prod_stats, by = "Prod")
sales_stats$Z <- (sales_stats$Uprice.x - sales_stats$Uprice.y[, "mean"]) /
                 sales_stats$Uprice.y[, "sd"]

sales_stats$FraudRisk <- ifelse(abs(sales_stats$Z) > 3, "Alto", "Baixo")
```

7 7. Modelagem Preditiva Semissupervisionada

Utilizamos o método de classificação semissupervisionada com o algoritmo Random Forest para extrapolar o conhecimento dos dados rotulados (‘ok’ e ‘fraud’) para as transações de status desconhecido. Esse modelo captura padrões complexos entre as variáveis disponíveis para estimar probabilidades de fraude, oferecendo uma previsão automatizada que contribui para a eficiência do processo de auditoria.

Treinamos o modelo com os dados rotulados e aplicamos as previsões nas observações não rotuladas, adicionando as previsões ao dataset para posterior análise.

```r
library(randomForest)

# Dados rotulados
train_data <- sales_stats[sales_stats$Insp %in% c("ok", "fraud"), ]
train_data$Insp <- factor(train_data$Insp)

# Dados não rotulados
test_data <- sales_stats[!sales_stats$Insp %in% c("ok", "fraud"), ]

# Modelo Random Forest
set.seed(123)
rf_model <- randomForest(Insp ~ Quant + Uprice.x, data = train_data)

# Previsões
preds <- predict(rf_model, newdata = test_data)

# Anexar resultados
test_data$Predito <- preds
head(test_data)
```

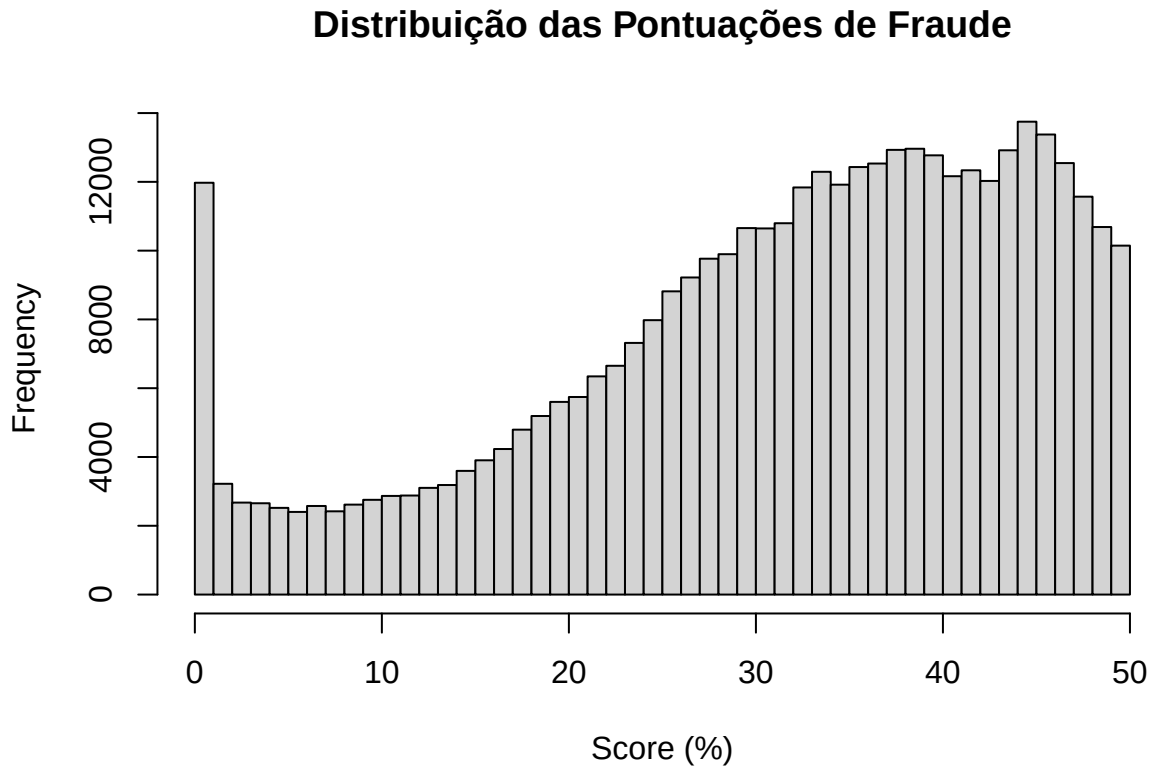
```
##   Prod    ID Quant   Val Insp Uprice.x Cluster Uprice.y.mean Uprice.y.sd
## 1   p1    v1   182  1665 unkn 9.148352       1     13.628542    7.870953
## 2   p1    v2  3072  8780 unkn 2.858073       1     13.628542    7.870953
## 3   p1    v3 20393 76990 unkn 3.775315       1     13.628542    7.870953
## 4   p1    v4   112  1100 unkn 9.821429       1     13.628542    7.870953
## 5   p1    v3  6164 20260 unkn 3.286827       1     13.628542    7.870953
## 6   p1 v4835  1722  7670 unkn 4.454123       1     13.628542    7.870953
##            Z FraudRisk Predito
## 1 -0.5692055     Baixo      ok
## 2 -1.3683817     Baixo      ok
## 3 -1.2518467     Baixo      ok
## 4 -0.4836915     Baixo      ok
## 5 -1.3139088     Baixo      ok
## 6 -1.1656045     Baixo      ok
```

8 8. Sistema de Pontuação de Fraude

Com base nos escores Z calculados anteriormente, atribuimos uma pontuação de fraude probabilística, convertendo o valor absoluto do escore Z em uma probabilidade estatística associada à anormalidade da transação. Essa pontuação facilita a triagem das transações mais suspeitas, possibilitando a criação de rankings para priorização das análises e ações corretivas.

Um histograma mostra a distribuição geral dessas pontuações no conjunto de dados.

```
sales_stats$FraudScore <- pnorm(abs(sales_stats$Z), lower.tail = FALSE) * 100

hist(sales_stats$FraudScore, breaks = 40,
     main = "Distribuição das Pontuações de Fraude",
     xlab = "Score (%)")
```




```
# Top 10 suspeitos
cols_exist <- intersect(c("ID", "Prod", "Uprice", "FraudScore"), names(sales_stats))
head(sales_stats[order(-sales_stats$FraudScore), cols_exist], 10)
```

```
##           ID  Prod FraudScore
## 124675 v2004 p1927   49.99989
## 100028 v4261 p1775   49.99981
## 266487  v549 p3273   49.99978
## 66208  v2179 p1549   49.99954
## 18210   v846 p1125   49.99950
```

```
## 342216 v5138 p4094   49.99948
## 342300 v5138 p4094   49.99948
## 342549 v5138 p4094   49.99948
## 342801 v5138 p4094   49.99948
## 175263 v5750 p2273   49.99941
```

9 9. Comparação com IA Generativa

A Inteligência Artificial generativa representa uma abordagem mais avançada e automatizada para detecção de fraudes, incorporando técnicas como Isolation Forest, Autoencoders e XGBoost, além de oferecer mecanismos de explicabilidade, como os valores SHAP para interpretação dos modelos.

Apesar da maior complexidade técnica e demanda computacional, esses métodos proporcionam melhor adaptação a padrões sofisticados e dinâmicos de fraude. Em contrapartida, o modelo estatístico tradicional oferece simplicidade, fácil interpretabilidade e menor custo operacional.

10 10. Conclusão

Este relatório apresentou uma análise detalhada de dados de vendas visando identificar possíveis fraudes usando técnicas combinadas de outlier, clustering e modelagem preditiva. A comparação com abordagens de IA generativa destacou as vantagens e limitações de cada método, evidenciando caminhos para futuros aprimoramentos, como inclusão de mais variáveis, integração de metodologias híbridas e monitoramento contínuo.