Uso e aplicação de análise preditiva como objeto de pesquisa voltado a uma base de dados específica, utilizando R

Rilson Figueiredo Miranda [1], **Rodrigo Moreira Fagundes** [1]

[1]Instituto de Matemática – Universidade Federal da Bahia (UFBA)
CEP.: 40.170-110 – Salvador – BA – Brasil

{rilsonfigueiredo,rmfagundes}@gmail.com

Abstract. *This paper aims to describe an exercise of the KDD process in a database provided by a company owning a delivery application of home food, in order to raise which factors are decisive for the rejection of an application, which makes it possible the development of metrics that allow the visualization of each class analyzed in an isolated setting..*

Resumo. *Este artigo visa descrever um exercício do processo de KDD em uma base de dados provido por uma empresa proprietária de um aplicativo de entrega de alimentos em domicílio, com o intuito de levantar quais fatores são determinantes para a recusa de um pedido, o que torna possível a elaboração de métricas que permitem a visualização de cada classe analisada dentro de um cenário isolado.*

1. Introdução

A criação de novas tecnologias transforma o homem e cria novos problemas, que demandam novas tecnologias, em um ciclo de retroalimentação, desde a invenção das primeiras ferramentas ([BURKE 1997]). Como demonstra o autor, a cada revolução essa engrenagem se acelera, reduzindo a flutuação - o tempo de demanda e entrada de novos componentes. Tal perspectiva pode ser observada no campo da informação.

O uso cada vez mais intensivo de sistemas de informação, em ambientes cada vez mais complexos, e a expansão do uso de novas tecnologias intensivas em dados, como a internet das coisas (IoT), tem elevado drasticamente o volume de dados disponíveis, em um ritmo maior do que a ampliação da capacidade para análise e construção de conhecimento capaz de promover ações ([JACOBS 2009]). Essa avalanche de informações exorbita a capacidade humana de visualizar, gerenciar e formular modelos de predição ([FAYYAD et al. 1996]).

Se, por um lado, essa evolução vulnera o homem pela exposição dessa limitação, como expõe [WURMAN 1993], sob uma outra ótica, trouxe a reboque conceitos, técnicas e soluções computacionais que permitem a expansão da nossa capacidade em lidar com esse volume de dados, sob a égide de KDD (Knowledge Discovery in Databases).

Este artigo visa empregar o processo de KDD sobre uma base de dados proprietária, de um aplicativo de entrega de alimentos em domicílio, objetivando descrever as etapas do processo e as impressões dos participantes.

Do ponto de vista da simulação, objetiva-se criar um modelo de predição para recusas de pedidos. Essa informação poderia indicar iniciativas para reduzir as perdas com pedidos não entregues.

O artigo está organizado conforme a sequência de KDD desenhada por [FAYYAD et al. 1996]. Na seção 2 são descritas as etapas de pré-processamento e transformação. Em seguida, a identificação de modelos através de algoritmos de classificação, na seção 3. Para fechar o processo, na seção 4, são apresentados os resultados, dentro da ótica da exploração realizada. Por fim, no último tópico, as conclusões finais dos pesquisadores.

Será disponibilizada ainda uma sessão com um endereço eletrônico onde poderão ser feito download da base de dados utilizada como base para este projeto, assim como os códigos fonte.

2. Pré-processamento

Em KDD, a etapa de pré-processamento compreende a aplicação de várias técnicas para captação, organização, tratamento e a preparação dos dados [GPEA 2014]. Para [CLÉSIO, Flávio 2015] este é um fator determinante a respeito do tempo que se gasta cara consolidar informações em formatos simples.

A base de dados fornecida no formato CSV (Comma-Separated Values) possuía, em princípio, as informações dispostas abaixo:

```
> names(df)
 [1] "DATA_PEDIDO"                "HORA_PEDIDO"                "STATUS_PEDIDO"
                  "DIA"                            "AVALIACAO"
 [6] "TIPO_PRODUTO"               "PRODUTO"                    "DETALHES_
     PRODUTO"          "COMIDA"                         "BEBIDA"
[11] "QTD"                       "V_UNITARIO"                 "V_ENTREGA"
                  "V_DESCONTO"                     "TOTAL_PEDIDO"
[16] "FORMA_PAGAMENTO"           "ESTABELECIMENTO"            "TIPO_
     ESTABELECIMENTO"    "BAIRRO_ESTABELECIMENTO"  "QTD_BAIRROS_ATENDIDOS"
[21] "NOME_USUARIO"              "DDD_USUARIO"                "OPERADORA_
     USUARIO"          "FACEBOOK"                       "BAIRRO_USUARIO"
[26] "GASTO_TOTAL_USUARIO"       "QTD_PEDIDOS_USUARIO"        "DATA_PEDIDO_
     ANTERIOR"    "PLATAFORMA"                     "DATA_CADASTRO"
```

Dentre os atributos, foi selecionado como classe de predição o STATUS_PEDIDO, que indica se fora entregue ou recusado. As recusas representam 7,19% dos pedidos feitos e 6,34% do valor monetário total. A capacidade de predizer recusas significa para as organizações a possibilidade de reduzir custos e de identificar oportunidades de melhoria em seus serviços.

2.1. Data Cleaning

[CODY and WOOD 2011] consideram o data cleaning como um dos primeiros e mais importantes passos para qualquer processamento de dados. Nessa etapa é possível verificar quais valores estão corretos ou em conformidade com algum conjunto de regras.

Em uma análise inicial, buscou-se reduzir a dimensionalidade da base em questão, de forma a manter apenas os atributos que se mostrassem relevantes para a classificação. Os critérios para remoção utilizados foram:

- Atributos descritivos, de alta granularidade e que podem sofrer adições a qualquer momento (PRODUTO, DETALHES_PRODUTO, ESTABELECIMENTO, NOME_USUARIO, BAIRRO_USUARIO, BAIRRO_ESTABELECIMENTO);

- Datas (DATA_PEDIDO, DATA_CADASTRO);
- Atributos sem massa crítica e sem uma substituição razoável (AVALIACAO);
- Valores agregados (GASTO_TOTAL_USUARIO, QTD_PEDIDOS_USUARIO);
- Informações julgadas não-influenciadoras de recusa do pedido (OPERADORA_USUARIO, FACEBOOK, DDD_USUARIO, QTD_BAIRROS_ATENDIDOS);
- Atributos que podem ser inferidos dos demais (V_UNITARIO, COMIDA). ...

```
> drops <- c("DATA_CADASTRO", "QTD_PEDIDOS_USUARIO", "GASTO_TOTAL_
    USUARIO",
+                "QTD_BAIRROS_ATENDIDOS", "DETALHES_PRODUTO", "AVALIACAO",
    "DATA_PEDIDO",
+                "PRODUTO", "COMIDA", "ESTABELECIMENTO", "OPERADORA_USUARIO
    ", "FACEBOOK",
              "NOME_USUARIO", "DDD_USUARIO", "BAIRRO_USUARIO", "BAIRRO_
                 ESTABELECIMENTO")
> df <- df[ , !(names(df) %in% drops)]
```

Destarte, restaram:

```
> names(df)
 [1] "HORA_PEDIDO"                "STATUS_PEDIDO"              "DIA"
                                "TIPO_PRODUTO"               "BEBIDA"
 [6] "QTD"                        "V_UNITARIO"                 "V_ENTREGA"
                          "V_DESCONTO"                 "TOTAL_PEDIDO"
[11] "FORMA_PAGAMENTO"           "TIPO_ESTABELECIMENTO"       "DATA_PEDIDO_
    ANTERIOR"     "PLATAFORMA"
```

2.2. Transformação

Transformação é a etapa do KDD que antecede a mineração dos dados, ela consiste em alterar os dados de modo que seja mais adequado para o objetivo que se planeja atingir [PRASS 2012]. Com isso, alguns atributos foram transformados visando melhorar a qualidade da predição.

- HORA_PEDIDO fora convertido em TURNO_PEDIDO, categorizado para os valores MANHA, TARDE, NOITE e MADRUGADA;
- DATA_PEDIDO_ANTERIOR tornou-se um indicador de PRIMEIRO_PEDIDO (valores YES/NO);
- O BAIRRO_USUARIO apresentava o mesmo valor para localidades claramente diferentes, com base no DDD_USUARIO. Inicialmente optou-se por acrescentar esta informação ao início do BAIRRO_USUARIO. No entanto, devido ao número excessivo de instâncias (123), as informações foram descartadas.

```
# Categorizando a data de pedido anterior como indicativo de se aquele
    pedido foi o primeiro
df$PRIMEIRO_PEDIDO <- 'NO'
df$PRIMEIRO_PEDIDO[df$DATA_PEDIDO_ANTERIOR == "NULL"] <- 'YES'
# Removendo a variavel original
df$PRIMEIRO_PEDIDO = NULL

# Categorizando o turno do pedido
df$TURNO_PEDIDO <- substr(df$HORA_PEDIDO, 0, 2)
```

```
df$TURNO_PEDIDO <- sub(':', '', df$TURNO_PEDIDO)
df$TURNO_PEDIDO[df$TURNO_PEDIDO >= 4 & df$TURNO_PEDIDO < 11] <- 'MANHA'
df$TURNO_PEDIDO[df$TURNO_PEDIDO >= 11 & df$TURNO_PEDIDO < 18] <- 'TARDE
    '
df$TURNO_PEDIDO[df$TURNO_PEDIDO >= 18 & df$TURNO_PEDIDO <= 23] <- '
    NOITE'
df$TURNO_PEDIDO[df$TURNO_PEDIDO >= 0 & df$TURNO_PEDIDO < 4] <- '
    MADRUGADA'
# Removendo a variavel original
df$HORA_PEDIDO <- NULL

# Usando V_ENTREGA como indicador de dist ncia
df$V_ENTREGA_FACTOR <- cut(df$V_ENTREGA,
                           breaks = c(-1, 0, 2, 4, 6, max(df$V_ENTREGA)
                               ),
                           labels = c("Sem custo", "Custos at  R$2,00"
                               , "Maior que R$2,00 e menor que R$4,00",
                                "Maior que R$4,00 e menor que R$6,00",
                               "Maior que R$6,00"))
df$V_ENTREGA <- NULL
```

Após as transformações, os atributos remanescentes foram:

```
> names(df)
 [1] "STATUS_PEDIDO"          "DIA"                    "TIPO_PRODUTO"
                "BEBIDA"                 "QTD"
 [6] "V_UNITARIO"             "V_DESCONTO"             "TOTAL_PEDIDO"
                "FORMA_PAGAMENTO"        "TIPO_ESTABELECIMENTO"
[11] "DATA_PEDIDO_ANTERIOR"   "PLATAFORMA"             "TURNO_PEDIDO"
                "V_ENTREGA_FACTOR"
```

3. Classificação

Uma vez que os dados presentes são, em sua maioria, categóricos, o rol de opções algoritmos de classificação se apresenta restrito. Optou-se por usar árvore de decisão, adequado para o tipo de conjunto de dados e cujo resultado visualmente se aproxima do eixo focal do estudo, que é identificar os fatores mais relevantes para a recusa de um pedido. Ora, se a árvore de decisão possui os nós ordenados conforme o maior ganho de informação, é razoável dizer que o percurso iniciado pela raiz indica os fatores em ordem decrescente de relevância.

Dentre os algoritmos de árvore de decisão, o C4.5 é consagrado pelo seu sucesso evolutivo e relativamente melhorado, comparado com outros relevantes, como o *ID3* [QUINLAN 1993]. Para o experimento foi usado o C5.0, que é uma extensão do C4.5.

Em um primeiro momento, segmentamos a base aleatoriament em 70% para treino e o restante para teste.

```
## C5.0
Sys.setlocale(locale="C")
ind <- sample(2, nrow(df), replace=T, prob=c(0.7,0.3))
# fit model
fit <- C5.0(STATUS_PEDIDO~., data=df[ind==1,], trials=10)
```

```
# summarize the fit
print(fit)
# make predictions
predictions <- predict(fit, df[ind==2,])
# summarize accuracy
table(predictions, df$STATUS_PEDIDO[ind==2])
```

Diante dos parâmetros para aplicação do modelo, obteve-se a seguinte avaliação de acurácia:

```
predictions Entregue Recusado
   Entregue      2848       205
   Recusado         2        12
```

O algoritmo classificou conforme esperado 2860 registros, dos 3067 selecionados para teste, o que representa pouco mais de 93,25% de acurácia. 205 predições de Entregue deveriam, de fato, ser marcados como Recusado, enquanto o algoritmo falhou apenas 2 vezes ao identificar pedidos entregues como recusados.

Muito embora a acurácia tenha sido animadora - e tenha se mantido sem variação significativa em múltiplas repetições (amostragem de testes aleatória) -, a aplicação do modelo de predição sobre a integralidade do dataframe não trouxe revelações para o objetivo de identificar os fatores que conduzem a uma recusa. Isso ocorreu em função da supremacia dos produtos entregues sobre os recusados, o que conduzia, para quase todos os conjuntos de atributos testados, apenas a uma raiz. À exceção, a inclusão dos atributos do tipo de produto e de estabelecimento geravam uma árvore de decisão com apenas esses atributos, mas, quando reduzidos os seus níveis categóricos, retornava-se ao problema original de único nó na árvore de decisão.

Em função desse revés, os dados voltaram para a fase de pré-processamento, para reduzir a supremacia de um valor de predição, sem, no entanto, corromper as informações. Foram removidos, para tal fim, todas as relações entre usuário e estabelecimento que não possuíram sequer uma recusa, restando a seguinte composição de STATUS_PEDIDO:

```
Entregue Recusado
     576       689
```

Ao tentar obter novamente a árvore de decisão, observou-se, conforme esperado, uma ordem de prevalência dos atributos selecionados em relação às chances de um produto ser recusado.

```
dtree <- rpart(STATUS_PEDIDO ~ V_ENTREGA_FACTOR + TURNO_PEDIDO + PEDIDO
    _ANTERIOR, data=df3, method="class")
fancyRpartPlot(dtree)
```

4. Resultados

As regras obtidas da árvore de decisão podem guiar a elaboração de iniciativas para mitigar as recusas ou apontar oportunidades de novas pesquisas para compreender melhor os fenômenos.

```
> dtree
```

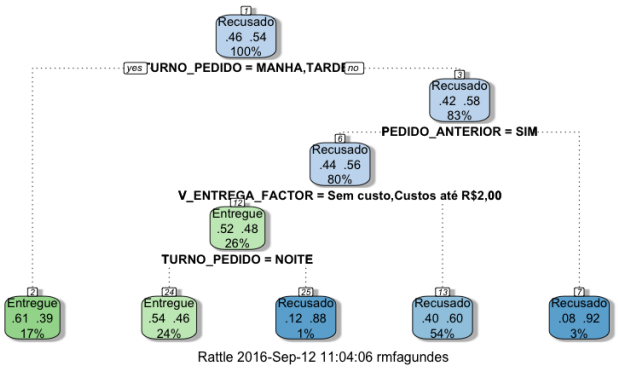

Figura 1. Árvore de decisão, com classe de predição o status do pedido, para relações cliente-estabelecimento que tenham resultado em ao menos uma recusa.

Fonte: Autoria Propria

```
n= 1265

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 1265 576 Recusado (0.45533597 0.54466403)
   2) TURNO_PEDIDO=MANHA,TARDE 215  84 Entregue (0.60930233 0.39069767)
       *
   3) TURNO_PEDIDO=MADRUGADA,NOITE 1050 445 Recusado (0.42380952
      0.57619048)
     6) PEDIDO_ANTERIOR=SIM 1012 442 Recusado (0.43675889 0.56324111)
      12) V_ENTREGA_FACTOR=Sem custo,Custos at<U+00E9> R$2,00 323 156
          Entregue (0.51702786 0.48297214)
        24) TURNO_PEDIDO=NOITE 307 142 Entregue (0.53745928 0.46254072)
            *
        25) TURNO_PEDIDO=MADRUGADA 16   2 Recusado (0.12500000
            0.87500000) *
      13) V_ENTREGA_FACTOR=Maior que R$2,00 e menor que R$4,00,Maior
          que R$4,00 e menor que R$6,00,Maior que R$6,00 689 275
          Recusado (0.39912917 0.60087083) *
     7) PEDIDO_ANTERIOR=N<U+00C3>O 38   3 Recusado (0.07894737
        0.92105263) *
```

Após a etapa de classificação, é possível visualizar algumas problematizações através do cruzamento de dados considerados relevantes na elaboração do modelo. Para exemplificar, no quadro à cima é identificado uma taxa de 60% de recusa nos pedidos realizados à noite ou na madrugada, feitos por pessoas que já efetuaram pedidos em outro momento e com uma taxa de entrega superior a R$2,00. Desse modo, é possível analisar o cenário e tentar inferir o fator motivante ao grande número de recusas.

5. Conclusão

Não existe analise de dados sem pré-processamento. Em toda elaboração de projeto existe a fase de planejamento que antecede a execução. Em Mineração de dados, esta etapa está diretamente ligada ao pré-processamento.

Apesar da base em questão ser simples, a disposição dos seus dados tornou a análise uma tarefa árdua, principalmente por apresentar um número muito superior de itens entregues em relação aos rejeitados, classe definida para predição. Neste item, a etapa de transformação foi crucial para que se pudesse ter uma amostra de dados mais relevante em relação o que se pretendia predizer, sem impactar na essência dos dados.

Como resultado de uma análise nem sempre o obtido é a resposta para a resolução dos problemas, mas pode de chegar a um ponto de partida para enxergá-lo de uma outra forma, ou simplesmente enxergá-lo.

A base de dados e os fontes utilizados como referência nesse artigo podem acessados através do endereço eletrônico: **https://github.com/RILSON/PGCOMP—Advanced-Topics-on-Database**.

Referências

BURKE, J. (1997). *The Axemaker's Gift: Technology's Capture and Control of Our Minds and Culture*. Tarcher.

CLÉSIO, Flávio (2015). **MLDB – MACHINE LEARNING DATABASE**. Disponível em https://mineracaodedados.wordpress.com/2015/09/28/mldb-machine-learning-database/. Acesso em: 12 de setembro de 2016.

CODY, R. E. and WOOD, R. J. (2011). **Data Cleaning**.Medical School. **Piscataway, NJ**.

FAYYAD, U., PIATETSKY-SHAPIRO, G., and SMYTH, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34.

GPEA (2014). **Pré-processamento em Data-Mining**. Disponível em http://www.din.uem.br/ gpea/linhas-de-pesquisa/mineracao-de-dados/pre-processamento/pre-processamento-em-data-mining/. Acesso em: 12 de setembro de 2016.

JACOBS, A. (2009). The Pathologies of Big Data. *Queue*, 7(6):10.

PRASS, F. S. (2012). **Um visão geral sobre as fases do Knowledge Discovery in Databases (KDD)**.

QUINLAN, J. R. (1993). **C4.5: Programs for machine learning**.San mateo Usa.

WURMAN, R. S. (1993). *Information anxiety*, volume 26.