INSTITUTO DE GESTÃO EM TECNOLOGIA DA INFORMAÇÃO

REVISÃO BIBLIOGRÁFICA DAS TAREFAS DE EXPLORAÇÃO E PRÉ-PROCESSAMENTO DE DADOS: SISTEMATIZAÇÃO DA PREPARAÇÃO DE DADOS PARA APLICAÇÃO DE ALGORITMOS DE *MACHINE LEARNING*

Arthur Pereira de Gouveia e Silva
Professor Especialista Alexandre Augusto de Carvalho Soares
MBA Ciência de Dados (*Big Data*)

Resumo

A aplicação dos algoritmos de *Machine Learning* deve ser precedida de tarefas de preparação dos dados, pois é um mito aplicar um algoritmo sobre dados brutos e esperar revelações importantes (LOHR, 2014 apud HEER). Outro fato para evidenciar a necessidade de tais tarefas é a existência de uma fase específica para preparação de dados na metodologia *Cross Industry Standard Process for Data Mining* (CRISP-DM). Além de necessárias, Alberto (2016) afirma que estas tarefas são as que mais consomem recursos computacionais, as que mais afetam o resultado final do processo, além de serem as etapas onde se gasta mais tempo nos projetos de *Data Mining*. Uma estimativa do custo financeiro do pré-processamento é 22.000 dólares por analista, de acordo com Haight (2016). Apesar da importância e da relevância das tarefas de preparação de dados no início dos projetos de *Data Mining*, tais tarefas não são sistematizadas. Coelho e Richert (2013) afirmam que a preparação de dados é mais arte do que ciência exata. Devido à importância e à falta de sistematização da preparação de dados surge a necessidade de organizar tais tarefas criando um fluxo de trabalho e consolidando as principais tarefas e ferramentas de pré-processamento de dados. Através de revisão e análise bibliográfica, este trabalho apresenta algumas técnicas e ferramentas utilizadas nas tarefas de preparação de dados. Após tal revisão, como conclusão do trabalho, uma proposta de fluxo de preparação e pré-processamento de dados é apresentada utilizando a linguagem Python e algumas bibliotecas disponíveis nessa linguagem.

Palavras-chave: Aprendizado de máquinas; Preparação de dados; Python; Pandas; Matplotlib

Abstract

The use of Machine Learning algorithms must be preceded by data preparation tasks since it's a myth that you will apply an algorithm over raw data and expect insights to pop up (LOHR, 2014 apud HEER). Another fact to highlight the need for those tasks is the existence of a specific phase for data preparation in Cross Industry Standard Process for Data Mining (CRISP-DM) methodology. Besides the necessity, according to Alberto (2016), those tasks are the ones that consume most computational resources, which most affect the result of the process besides being the steps where most time is spent on Data Mining projects. An estimate by Haight (2016) is that the preprocessing financial cost is 22,000 dollars per analyst. Despite of the importance and relevance of the data preparation tasks in the beginning of Data Mining projects, such tasks aren't systematized.

Coelho and Richert (2013) say that data preparation is more art than a precise science. Because of the importance and the lack of systematization of data preparation arises the need to systematize those tasks through creating a workflow and consolidating the main tasks and tools used on preprocessing tasks. This paper presents some techniques and activities of data preparation through a bibliographical study. After this study, as a conclusion of this work, a proposal for a data preparation and pre-processing workflow is presented using the Python language and some libraries available for it.

Keywords: Machine Learning; Data preparation; Python; Pandas; Matplotlib

1. INTRODUÇÃO

O volume de dados produzidos nos últimos dois anos, de acordo com Gallo (2016), é equivalente a 90% dos dados produzidos em toda a história da humanidade. Para gerar valor para os negócios se faz cada vez mais necessário acessar, compreender e analisar todo este volume de dados. Mas apenas analisar os dados armazenados a fim de entender o que aconteceu não é mais suficiente; é fundamental que as empresas compreendam o passado a fim de predizer o futuro.

Desta forma surgiu a necessidade do uso de algoritmos de aprendizado de máquina ou *Machine Learning* nas quais o computador “aprende” através dos dados históricos e, com certo grau de incerteza, prevê o que irá acontecer.

Entretanto, os algoritmos de *Machine Learning* não são à prova de dados de baixa qualidade – valores faltantes, constantes, correlacionados etc. – além de possuírem pressupostos sobre o tipo de dados que podem analisar – categóricos, numéricos, ordinais etc. Isso ilustra o que Lohr (2014 apud HEER) quis dizer ao afirmar que não podemos esperar bons resultados ao aplicar inadvertidamente algoritmos de aprendizado de máquina a dados brutos.

A fim de extrair verdadeiro valor dos dados e maximizar a capacidade de predição dos algoritmos de aprendizado de máquina é necessário trabalhar os dados brutos colocando-os em um formato aderente aos pressupostos dos diversos modelos. As

atividades de preparação, transformação, visualização, agregação e adequação dos dados são chamadas de pré-processamento, preparação dos dados ou *data wrangling*.

Uma pesquisa da *CrowdFlower* (2016) mostra que 60% dos cientistas de dados afirmam que gastam a maior parte do seu tempo nas tarefas de limpeza e organização dos dados e 57% dos pesquisados dizem que essas atividades são a parte menos prazerosa da ciência de dados. Alberto (2016) afirma ainda que estas tarefas também são as que mais consomem recursos computacionais.

Surge, então, um problema: como organizar e sistematizar as ferramentas, técnicas e tarefas de preparação de dados para que o cientista de dados seja mais produtivo, organizado e científico ao executá-las?

Diante do exposto, o objetivo deste trabalho é realizar uma revisão bibliográfica das ferramentas e técnicas utilizadas na preparação de dados e propor uma rota sistematizada para aplicação de tais técnicas de forma eficiente, pois falta na literatura atual uma sistematização da etapa de preparação dos dados. Como afirmam Coelho e Richert (2013, tradução nossa) “quando se precisa responder questões como a manipulação de valores inválidos ou faltantes, você verá que isso é mais arte do que ciência exata”. Além do fato de que a falta de organização na realização do trabalho de ciência de dados pode levar à ineficiência e à diminuição da qualidade dos modelos pelo uso de dados inadequados. Para alcançar tal objetivo, um fluxo de preparação de dados é proposto.

Este trabalho está estruturado com a apresentação da metodologia no item 2, a revisão da literatura no item 3 onde são apresentados os principais conceitos trabalhados ao longo do texto, a apresentação da pesquisa no item 4 onde as ferramentas e tarefas de preparação de dados são apresentadas, a discussão dos resultados no item 5 onde todo o conhecimento gerado ao longo do texto é consolidado com a apresentação do fluxo de preparação de dados, no item 6 são realizadas as considerações finais sobre o texto, em seguida as referências utilizadas são apresentadas e, finalmente, no APÊNDICE A é

mostrado um código em Python que realiza todas as tarefas de preparação de dados em uma base de exemplo.

2. METODOLOGIA

Este trabalho apresenta as ferramentas e técnicas de preparação, visualização e pré-processamento de dados através de uma pesquisa de natureza aplicada pois busca organizar o estado da arte do conhecimento para a resolução de problemas práticos (GIL, 2002).

A abordagem de acordo com o problema de pesquisa foi qualitativa. A pesquisa qualitativa, de acordo com Gil (2002), depende de vários fatores como a natureza dos dados que, no caso desta pesquisa, tratam-se de textos já publicados. Quanto aos objetivos, trata-se de uma pesquisa exploratória. Ferreira (2016) afirma que a pesquisa exploratória objetiva proporcionar maior familiaridade com um problema e envolve levantamento bibliográfico.

Do ponto de vista dos procedimentos técnicos o trabalho apresenta uma pesquisa bibliográfica que consiste em um apanhado geral de trabalhos realizados anteriormente (MARKONI e LAKATOS, 2003). A pesquisa foi realizada através da análise de livros, artigos, periódicos e textos da internet publicados seguida da consolidação do conhecimento com a finalidade de:

- Apresentar uma metodologia de Mineração de Dados
- Identificar as etapas de preparação e pré-processamento de dados na metodologia de Mineração de Dados.
- Identificar as ferramentas e técnicas mais usuais nessas etapas
- Produzir um fluxo de utilização das ferramentas e técnicas de preparação de dados

3. REVISÃO DE LITERATURA

Nesta seção são apresentados diversos conceitos fundamentais para a compreensão do trabalho.

3.1 *Machine Learning*

O conceito de *Machine Learning* (também chamado de *Data Mining*, ou Análise Preditiva) não é recente. North (2012, p.5) afirma que as raízes da mineração de dados podem ser encontradas no fim dos anos 1980. De acordo com Coelho e Richert (2015, p.2) o sucesso deste campo nos últimos anos pode ser atribuído ao uso pragmático de técnicas sólidas de outros campos como a estatística.

O objetivo do aprendizado de máquinas é ensinar às máquinas (*softwares*) como executar tarefas lhes apresentando alguns exemplos de como fazer ou de como não fazer as tarefas. Harrington (2012, p.5) afirma que *Machine Learning* é a interseção entre ciência da computação, engenharia e estatística e pode ser aplicado em várias áreas. Exemplos típicos de uso de *Machine Learning* é a detecção de *spam* em e-mails, reconhecimento de objetos em imagens, compreensão de linguagem natural e recomendação de produtos.

3.2 Estatística Descritiva

Segundo Triola (1999, p.144) “a Estatística Descritiva tem por objetivo resumir ou descrever características importantes de dados populacionais conhecidos” e consiste em descrever, explorar e comparar dados.

A Estatística Descritiva atinge esse objetivo através do uso de tabelas, gráficos e medidas como média, mediana e desvio padrão.

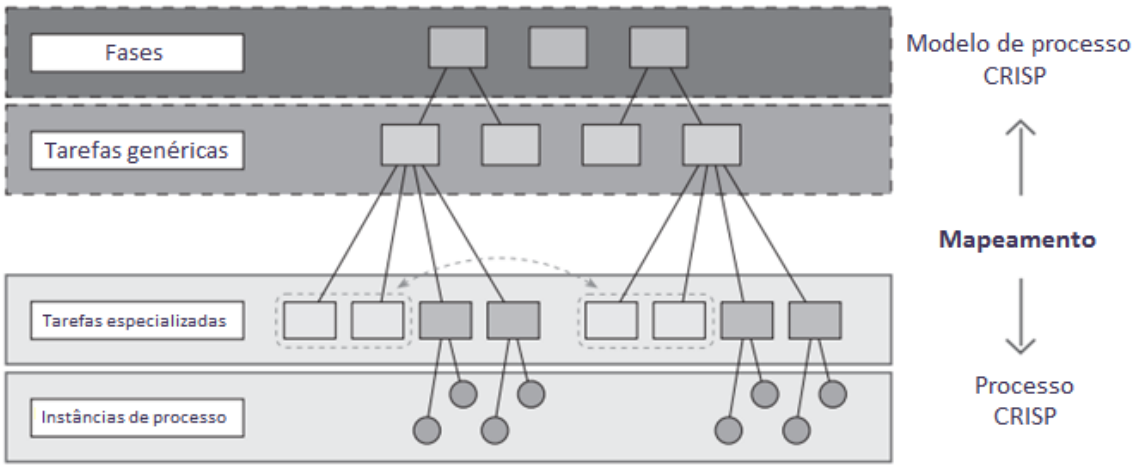
Ela se diferencia da Estatística Inferencial no sentido de que esta, através dos dados amostrais, infere características da população enquanto aquela apenas descreve os dados amostrais.

3.3 Cross Industry Standard Process for Data Mining (CRISP-DM)

O CRISP-DM, como colocado por Chapman *et al* (2000), é uma referência de metodologia para *Data Mining* e *Machine Learning* concebida em 1996 por três empresas: DaimlerChrysler, SPSS e NCR. À época o interesse por *Data Mining* estava crescendo e, em 1997, foi formado um consórcio com o objetivo de gerar um modelo padrão, não-proprietário e disponível livremente. O objetivo do modelo foi fornecer um caminho para os novos praticantes e demonstrar que o assunto já estava maduro o suficiente para ser usado como parte chave dos processos de negócios.

A metodologia CRISP-DM apresenta uma quebra hierárquica entre o modelo de processo CRISP e o processo CRISP através de quatro níveis de abstração: fase, tarefa genérica, tarefa especializada e instância de processo (Figura 1).

Figura 1 - A quebra em quatro níveis da metodologia CRISP-DM



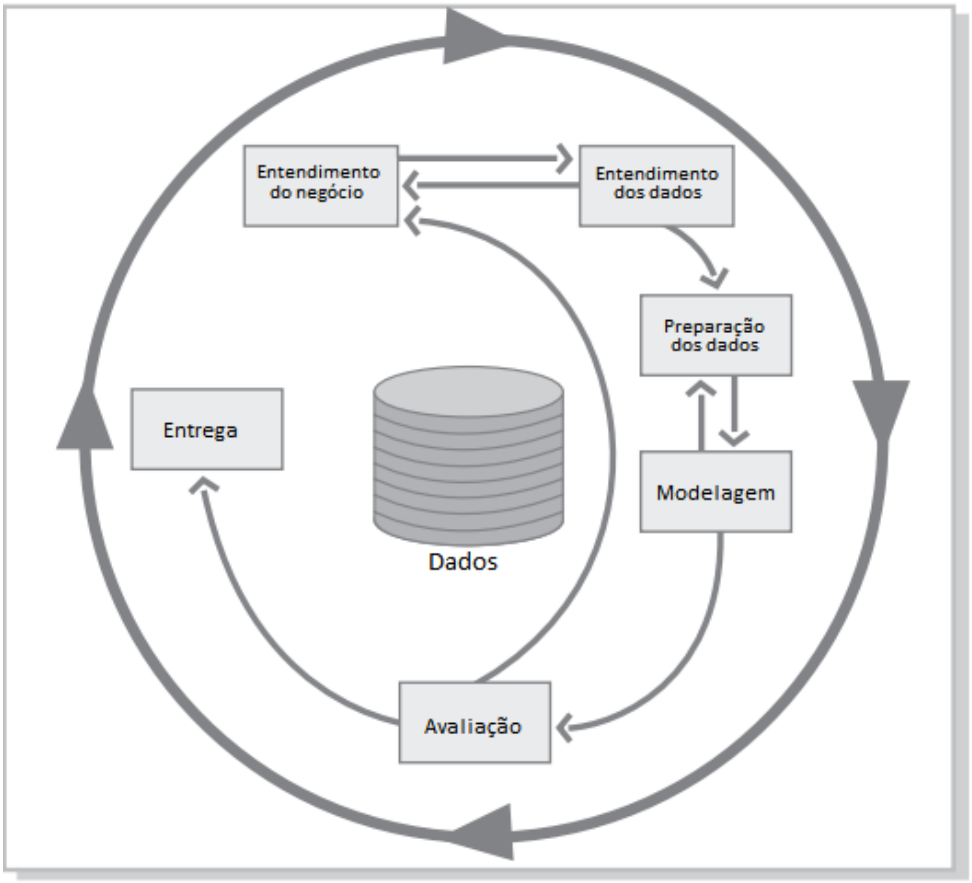
Hierarquia da metodologia CRISP-DM e sua quebra em Modelo de processo CRISP e Processo CRISP.
Fonte: Chapman *et.al*, 2000 p. 6

O modelo de referência da metodologia CRISP-DM afirma que o ciclo de vida de um projeto de *Data Mining* consiste de seis fases:

- Entendimento do negócio
- Entendimento dos dados
- Preparação dos dados
- Modelagem
- Avaliação
- Entrega

Essas seis fases são apresentadas na Figura 2.

Figura 2 - Fases do modelo de referência do CRISP-DM



Apresentação do ciclo do CRISP-DM e suas seis fases

Fonte: Chapman *et.al*, 2000 p. 6

Observa-se que as fases de preparação dos dados e modelagem são iterativas. Um resultado não satisfatório na fase de modelagem, ou alguma indicação de erro por parte do algoritmo fazem com que novas tarefas de preparação dos dados tenham que ser executadas.

3.4 Preparação de dados

Famili *et al* (1997) define o pré-processamento de dados como uma transformação T que transforma os vetores de dados do mundo real X_{ik} em um novo conjunto de vetores de dados Y_{ij}.

$$Y_{ij} = T(X_{ik})$$

(1)

de modo que 1. Y_{ij} preserve o valor das informações em X_{ik}, 2. Y_{ij} elimina ao menos um dos problemas de X_{ik} e 3. Y_{ij} é mais útil para a modelagem que X_{ik}. Na fórmula (1) temos que:

$i = 1, ..., n$ onde n = número de amostras ou quantidade de linhas na tabela,

$j = 1, ..., m$ onde m = número de atributos ou colunas na tabela após o pré-processamento,

$k = 1, ..., l$ onde l = número de atributos ou colunas na tabela antes do pré-processamento,

e geralmente $m \neq l$

Assim sendo, a preparação de dados consiste em deixar os dados brutos em um formato utilizável pelos algoritmos de *Machine Learning*. No contexto do presente trabalho a preparação de dados envolve algumas tarefas da segunda e terceira fases do CRISP-DM: as fases de entendimento dos dados e de preparação de dados. Estas são as tarefas de preparação dos dados segundo Chapman *et al.* (2000)

- Descrever os dados
    - Examinar superficialmente as propriedades dos dados
- Explorar os dados
    - Manipular os dados através de técnicas de Estatística Descritiva a fim de obter maior conhecimento sobre os dados antes de aplica-los aos algoritmos de *Machine Learning*.

- Verificar a qualidade dos dados
    - Examinar a qualidade dos dados sob alguns pontos de vista como: os dados estão completos? São corretos ou contém erros? Se contém erros, o quão comum eles são? Existem valores faltantes? Em caso positivo, como eles são representados, onde ocorrem, qual sua frequência?
- Selecionar dados
    - Esta tarefa consiste em decidir quais dados serão utilizados na modelagem. Essa decisão pode ser feita através da verificação da relevância dos dados para a análise, a qualidade dos dados, limitações técnicas quanto ao tipo de dados e à quantidade dos mesmos. A seleção de dados ocorre tanto na escolha dos atributos (colunas) quanto das amostras (linhas).
- Limpar os dados
    - Aumentar a qualidade dos dados ao nível requerido pelo modelo de *Machine Learning*. Tal aumento pode ocorrer através de subconjuntos dos dados que estejam mais limpos, ou a definição de valores padrão para dados faltantes. Ou até mesmo através de técnicas mais sofisticadas como a previsão de valores faltantes através de modelagem.

4. APRESENTAÇÃO DA PESQUISA

Nessa sessão são apresentadas algumas técnicas e ferramentas para realizar as tarefas de preparação dos dados.

4.1 Descrever os dados

A descrição dos dados é uma tarefa importante e deve ser a primeira ação a ser feita após acessar uma fonte de dados. Essa descrição inicial permite ao cientista de dados se familiarizar com os dados e ter uma noção da qualidade dos dados. A descrição dos

dados nessa fase é superficial e serve apenas obter uma ideia geral da propriedade dos dados. As principais tarefas da descrição de dados são:

- Apresentar as dimensões da base de dados (Quantidade de atributos e de amostras)
- Observar algumas linhas da fonte de dados
- Verificar a quantidade de dados faltantes
- Verificar os tipos de dados.

4.2 Explorar os dados

Nessa fase são utilizadas ferramentas da estatística descritiva e da análise exploratória de dados a fim de fornecer ao cientista de dados um conhecimento mais profundo dos dados, sua ordem de grandeza, frequência de ocorrência, nível de cardinalidade etc.
As principais ferramentas e técnicas dessa fase são:

- Histograma
- *Boxplot*
- Gráfico de dispersão
- Matriz de correlação
- Gráficos de barras
- Tabela de frequências
- Resumo de cinco números

4.3 Verificar a qualidade dos dados

A descrição dos dados e a exploração inicial já mostraram os principais problemas de qualidade dos dados. Dados faltantes já foram observados na descrição dos dados e a análise das tabelas de frequência, do histograma ou do *boxplot* podem indicar dados incorretos. Por exemplo uma variável chamada **sexo** que apresente a tabela de frequências mostrada no Quadro 1 é um indício de baixa qualidade dos dados.

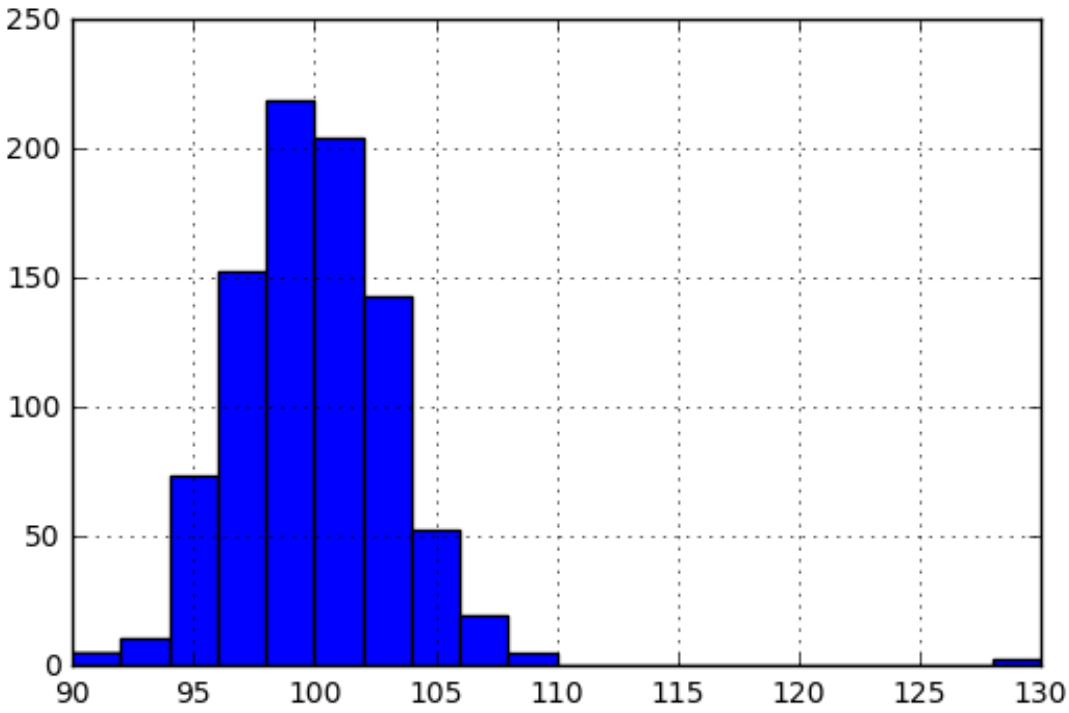
Quadro 1: Exemplo de uma tabela de frequências indicando baixa qualidade de dados

Sexo	Frequência
F	2842
M	3224
S	8
N/I	1

Fonte: O autor, 2017

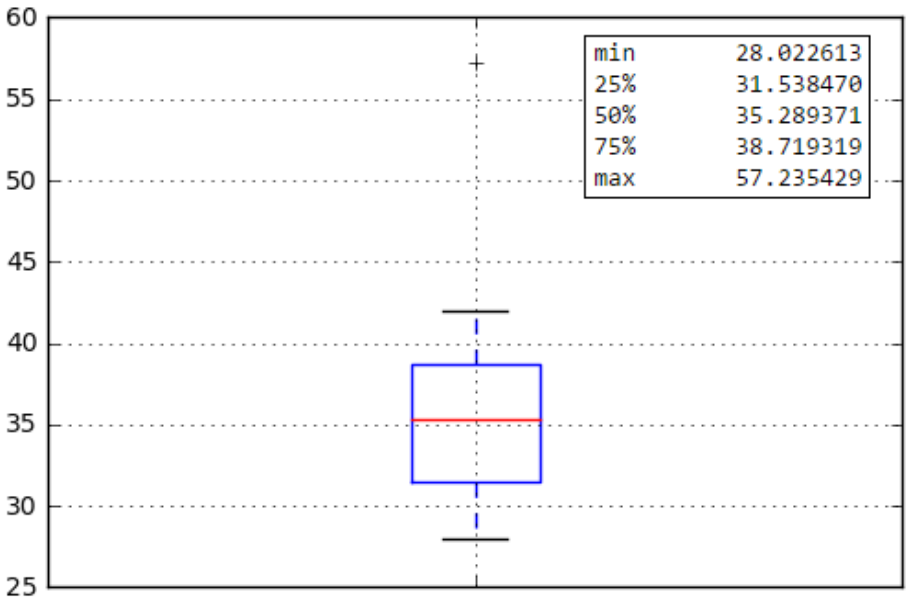
Dados cujo histograma seja o exibido na Figura 3 ou cujo *boxplot* e resumo de cinco números sejam o da Figura 4 também apresentam problemas de baixa qualidade.

Figura 3: Exemplo de histograma evidenciando dados de baixa qualidade



Histograma mostrando dados de baixa qualidade. A maior parte dos dados se distribui entre 90 e 110 mas existem dados extremos ao redor de 130.
Fonte: O autor, 2017

Figura 4: *Boxplot* e resumo de cinco números como exemplo de baixa qualidade dos dados



O *boxplot* representa de forma gráfica o resumo de cinco números e o ponto isolado de valor 57,2 aproximadamente é um valor extremo ou *outlier*.
Fonte: O autor, 2017

Dados com gráficos como os apresentados nas Figuras 3 ou 4 permitem observar visualmente a baixa qualidade em virtude de valores extremos. Tanto o valor 130 na Figura 3 quanto o ponto em torno de 57 na Figura 4 são dados que aparentemente não fazem parte do conjunto de dados.

4.4 Selecionar dados

Essa tarefa envolve selecionar, entre os dados disponíveis, quais serão utilizados na fase de modelagem. A seleção de dados envolve escolher atributos ou mesmo amostras. Algumas regras para seleção das variáveis são:

- Qualidade dos dados: Amostras com mais de 20% de atributos faltantes devem ser eliminadas (FAMILI, *et al*. 1997)
- Atributos fortemente correlacionados com outros não devem ser mantidos
- Atributos de data podem ser decompostos em dia, mês, ano, dia da semana, bimestre, trimestre, quinzena etc.

4.5 Limpar os dados

A tarefa de limpeza dos dados é a última etapa do pré-processamento e ao seu término os dados estão prontos para serem fornecidos ao algoritmo de modelagem. Algumas tarefas são:

- Preencher valores faltantes: valores faltantes podem ser tratados substituindo-os pela média ou moda dos valores presentes ou até mesmo tentar prever os valores faltantes através de algum algoritmo de *Machine Learning*.
- Corrigir dados incorretos: Se foi possível identificar a causa dos erros nos dados e foram tomadas medidas para que tais erros não ocorram novamente os dados errados podem ser corrigidos se possível. Ou então serem tratados como dados faltantes.
- Atributos do tipo *string* devem ser padronizadas para maiúscula ou minúscula e espaços desnecessários devem ser removidos
- Os dados devem ter seu tipo alterado para poderem ser aplicados à fase de modelagem. Caso opte-se por um algoritmo que só aceite dados numéricos e existam variáveis textuais na base, os dados devem ser codificados em inteiro ou transformados em *dummy*. A Figura 5 ilustra a transformação de um atributo transformado em *dummy*.

Figura 5: Exemplo de transformação de uma variável categórica em *dummy*.

cor
Azul
Azul
Amarelo
Verde
Vermelho
Preto
Verde
Preto
Vermelho
Azul
Amarelo
Azul

cor_Azul	cor_Verde	cor_Amarelo	cor_Vermelho	cor_Preto
1	0	0	0	0
1	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	0	0	1	0
0	0	0	0	1
0	1	0	0	0
0	0	0	0	1
0	0	0	1	0
1	0	0	0	0
0	0	1	0	0
1	0	0	0	0

Transformação de uma variável textual em várias colunas que assumem apenas valores 0 ou 1.
Fonte: O autor, 2017

- Acertar a escala das variáveis numéricas caso isso seja necessário para o algoritmo. As ações mais comuns são a normalização z onde cada dado é subtraído da média e dividido pelo desvio padrão. Outra forma é padronizar todos os dados de 0 a 1.

5. DISCUSSÃO DOS RESULTADOS

Através da análise das tarefas e atividades relacionadas à preparação e pré-processamento dos dados é possível sistematizar uma sequência de tarefas e de ferramentas que são utilizadas no pré-processamento dos dados. Tais tarefas se iniciam após a coleta dos dados e seguem até que os mesmos estejam prontos para a inserção no algoritmo de *Machine Learning*.

É importante salientar que o pré-processamento não é uma tarefa linear e que, quando concluída, a fase de modelagem não necessariamente será executada sem maiores problemas. É possível que o resultado do modelo não atenda aos requisitos do negócio e isso pode envolver a coleta de novos dados e uma nova fase de pré-processamento; também é possível que o algoritmo apresente alguma mensagem de erro sobre um problema não percebido na análise dos dados forçando o cientista de dados a voltar à fase de preparação.

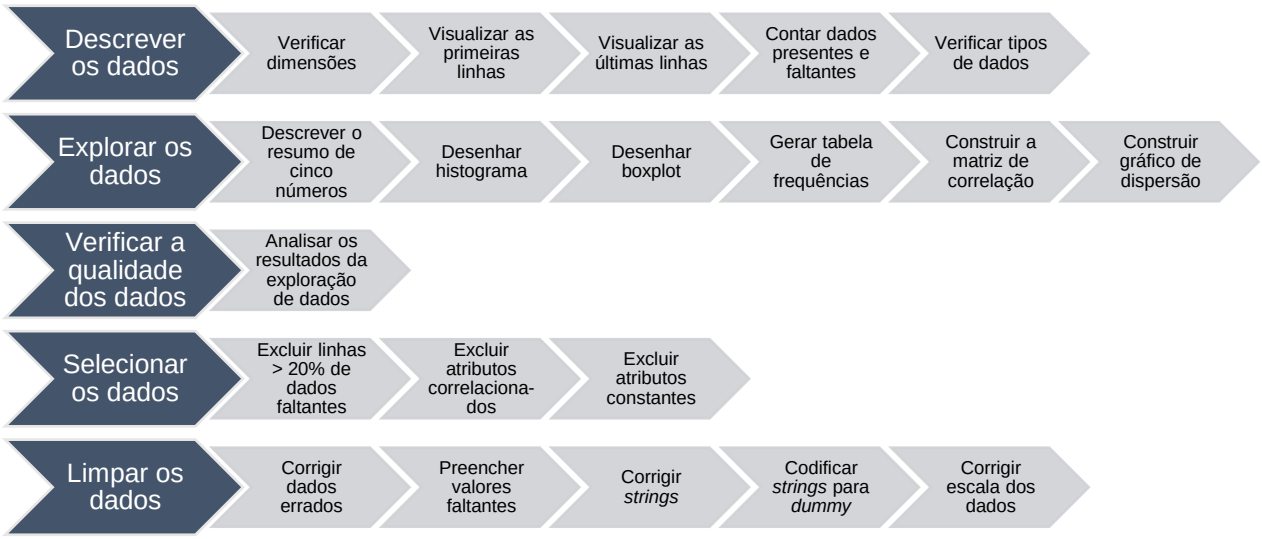
As tarefas sugeridas como um fluxo para o pré-processamento são as seguintes:

1. Descrever os dados
    a. Verificar as dimensões da base de dados – Número de linhas e colunas
    b. Visualizar as primeiras linhas da base
    c. Visualizar as últimas linhas da base
    d. Contar a quantidade de dados presentes e dados faltantes
    e. Verificar os tipos de dados
2. Explorar os dados
    a. Descrever os dados numéricos através do seu resumo de cinco números

   b. Desenhar o histograma dos dados contínuos
   c. Se necessário, desenhar o *boxplot* dos dados contínuos
   d. Construir a tabela de frequências de dados discretos. Se necessário desenhar um gráfico de barras a partir da tabela de frequências
   e. Construir a matriz de correlação entre os dados numéricos
   f. Construir um diagrama de dispersão para as maiores correlações
3. Verificar a qualidade dos dados
   a. Analisar o resultado da tarefa 2 e diagnosticar a qualidade dos dados
4. Selecionar os dados
   a. Excluir linhas com mais de 20% de dados faltantes
   b. Excluir atributos muito correlacionados a outros
   c. Excluir atributos constantes
5. Limpar os dados
   a. Corrigir dados errados ou transformá-los em valores vazios
   b. Preencher os valores faltantes que eventualmente restarem
   c. Corrigir variáveis do tipo *string* eliminando espaços desnecessários e corrigindo o caso para maiúsculas ou minúsculas.
   d. Se for necessário codificar os atributos *string* para inteiro ou transformá-los em *dummy*.
   e. Se necessário corrigir a escala dos dados padronizando-os através da padronização z ou colocando-os na escala de 0 a 1

A execução das tarefas listadas acima organiza o trabalho do cientista de dados através de uma metodologia clara e objetiva. Tais tarefas são estruturadas em forma de fluxo na Figura 6

Figura 6: Fluxo das tarefas de pré-processamento de dados



Apresentação do fluxo sistematizado de pré-processamento de dados
Fonte: O autor, 2017

Apresentaram-se várias técnicas e ferramentas encontradas de forma dispersa na bibliografia.

Observou-se que a execução das tarefas de preparação de dados através da metodologia proposta no trabalho organiza e sistematiza o trabalho do cientista de dados antes da aplicação da metodologia.

6. CONSIDERAÇÕES FINAIS

O pré-processamento de dados é parte importante do trabalho de um cientista de dados e fundamental para o mesmo obter bons resultados em seus modelos. Entretanto a falta de organização das tarefas de preparação de dados pode gerar perda de produtividade entre idas e vindas entre a aplicação do modelo, avaliação dos resultados e nova preparação de dados.

Conclui-se, então, sobre a necessidade de um método lógico e organizado de pré-processamento de dados. A bibliografia, além de nos mostrar a importância de tal fato, já

nos mostra, de forma pouco aprofundada e sem citar técnicas ou ferramentas, uma sequência de passos para a realização do tratamento de dados.

O estudo atual alcança seu objetivo de organizar as técnicas, ferramentas e tarefas em um fluxo sistematizado. Entretanto, o estudo não analisa se há ganho de tempo ao aplicar a metodologia proposta em um projeto real de análise de dados e não possui tal objetivo.

Sugere-se para um estudo futuro a avaliação do ganho de produtividade de um cientista de dados ao aplicar a metodologia apresentada.

REFERÊNCIAS

ALBERTO, Bruno Lambertucci Araújo. **Descoberta de conhecimento com Big Data Analytics**. Belo Horizonte: IGTI, 2016

CHAPMAN, P. *et al*, **CRISP-DM 1.0: Step-by-step data mining guide**, 2000 Disponível em: <https://www.the-modeling-agency.com/crisp-dm.pdf>. Data de acesso 05 feb. 2017

COELHO, L. P.; RICHERT, W **Building Machine Learning Systems with Python**. Birmingham: Packt Publishing, 2013

CrowdFlower. **Data Science Report**, 2016 Disponível em: <http://visit.crowdflower.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf>. Data de acesso: 15 abr. 2017

FAMILI, A.; SHEN, W. M.; WEBER, R.; SIMOUDIS, E. Data Preprocessing and Intelligent Data Analysis **International Journal on Intelligent Data Analysis**, Lansdale PA EUA, vol. 1, pp. 3-23, 1997

FERREIRA, Emanuele Berenice Marques. **Metodologia para Projeto Aplicado**. Belo Horizonte: IGTI, 2016

GALLO, João Guilherme. **Fundamentos de Big Data**. Belo Horizonte: IGTI, 2016

GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo, SP: Atlas, 2002.

HARRINGTON, P. **Machine Learning in Action**. Shelter Island, NY: Manning Publications co., 2012

LOHR, Steve. **For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights**, 2014. Disponível em: <https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html>. Data de acesso: 27 mar. 2017.

MARCONI, M. A.; LAKATOS, E. M. **Metodologia do Trabalho Científico**. 5. ed. São Paulo, SP: Atlas, 2003.

NORTH, M. A. **Data Mining for the Masses**, Global Text Project, 2012

TRIOLA, Mário F. **Introdução à Estatística**. 7. ed. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora, 1999

APÊNDICE A – Exemplo da aplicação da metodologia de preparação de dados

```
# coding: utf-8

# REVISÃO BIBLIOGRÁFICA DAS TAREFAS DE EXPLORAÇÃO E PRÉ-PROCESSAMENTO DE DADOS:
# SISTEMATIZAÇÃO DA PREPARAÇÃO DE DADOS PARA APLICAÇÃO DE ALGORITMOS
# MACHINE LEARNING
#
#                                                Arthur Pereira de Gouveia e Silva
#                   Professor Especialista Alexandre Augusto de Carvalho Soares
#                                                MBA Ciência de Dados (Big Data)


# ### Importação das bilbiotecas necessárias ###

import pandas as pd
from sklearn.linear_model import LinearRegression


# ### Leitura da fonte de dados do Titanic ###
# Disponível em https://www.kaggle.com/c/titanic/data

# Assume que os dados estão no diretório atual, no subdiretório datasets,
# no arquivo train.csv
titanic = pd.read_csv("datasets/train.csv")


# ### 1. Descrever os dados ###
print("Tarefa 1: Descvrever os dados")
print()

# Verificar as dimensões
print("Dimensões da base de dados")
print(titanic.shape)
print()

# Visualizar as primeiras linhas
print("Primeiras linhas")
print(titanic.head())
print()

# Visualizar as últimas linhas
print("Últimas linhas")
print(titanic.tail())
print()

# Contar os dados presentes por artibuto
print("Quantidade de dados por atributo")
print(titanic.notnull().apply(lambda x: sum(x)))
print()

# Contar os dados faltantes por atributo
```

```
print("Quantidade de dados faltantes por atributo")
print(titanic.isnull().apply(lambda x: sum(x)))
print()

# Verificar os tipos de dados
print("Tipos de dados")
print(titanic.dtypes)
print()

# Uma forma mais simples de chegar às mesmas informações acima
# print(titanic.info())


# ### 2. Explorar os dados ###
print("Tarefa 2: Explorar os dados")
print()
# Uma transformação prévia para que PassengerId, Pclass e Survived
# não sejam confundidos com variáveis numéricas
titanic.PassengerId = pd.Categorical(titanic.PassengerId)
titanic.Pclass = pd.Categorical(titanic.Pclass)
titanic.Survived = pd.Categorical(titanic.Survived)

# Descrever o resumo de cinco números
# A função describe traz também a quantidade de dados, a média e o desv padrão
print("Resumo de cinco números (mais count, média e stdev)")
print(titanic.describe())

# Desenhar histograma
_ = titanic.hist(figsize=(10, 10))

# Desenhar boxplot
_ = titanic.boxplot(figsize=(10, 10))


# Gerar tabela de frequências
print("Tabelas de frequência")
for coluna in titanic.columns:
    if titanic[coluna].dtype.name in ['category', 'object']:
        print(coluna)
        print(titanic[coluna].value_counts())
        print()


# Construir a matriz de correlação
matriz_correlação = titanic.corr()
print("Matriz de correlação")
print(matriz_correlação)

# Construir os gráficos de dispersão
_ = pd.scatter_matrix(titanic, diagonal='kde', figsize=(10, 10))


# ### 3. Verificar a qualidade dos dados ###
print("Tarefa 3: Verificar a qualidade dos dados")
```

```
msg = """
Analisando o resultado das tarefas anteriores os dados parecem ter os seguintes
problemas de qualidade:
1. Valores faltantes em Age, Cabin e Embarked
2. Possíveis valores incorretos em Fare; valores acima de 500"""
print(msg)
print()

# ### 4. Selecionar os dados
print("Tarefa 4: Selecionar os dados")

# Verificar se alguma linha tem mais de 20% de dados faltantes
print("Removendo linhas com mais de 20% de dados faltantes")
# Quantidade de colunas
num_cols = len(titanic.columns)
# Lista de valores bool indicando se a linha tem + de 20% de valores faltantes
excesso_missing = titanic.apply(lambda x: sum(x.isnull())/num_cols,
                                axis=1) > 0.2

# Quantidade de linhas antes da remoção
print('Quantidade de linhas antes da remoção')
print(len(titanic), '\n')

# Quantidade de linhas com mais de 20% de valores faltantes
print('Quantidade de linhas com mais de 20% de valores faltantes')
print(sum(excesso_missing), '\n')

# Mantém apenas as linhas com menos de 20% de valores faltantes
titanic = titanic.loc[~excesso_missing, :]

# Quantidade de linhas após a remoção
print('Quantidade de linhas após a remoção')
print(len(titanic), '\n')


def remove_altas_correlações(df, threshold):
    """
    Remove a coluna que apresenta correlação acima de threshold com a maior
    quantidade de outras colunas

    df: pandas dataframe a ser avaliado

    threshold: limite de correlação aceito. Correlações acima de threshold e
    abaixo de -threshold serão removidas

    Retorna uma tupla com o dataframe alterado e a coluna removida
    """
    from collections import Counter

    matriz_correlação = df.corr()
    corr = matriz_correlação.stack()
    corr = corr[corr.index.get_level_values(0) !=
                corr.index.get_level_values(1)]
    high_corr = corr[(corr > threshold) |
                     (corr < -threshold)].drop_duplicates()
```

```
    colunas = list(high_corr.index.get_level_values(0)).extend(
              list(high_corr.index.get_level_values(1)))
    if colunas:
        coluna_a_remover = Counter(colunas).most_common(1)[0][0]
        return df[df.columns.difference([colunas_a_remover])], coluna_a_remover
    else:
        return df, None


# Excluir atributos muito correlacionados a outros
print("Excluindo atibutos com |correlação| > 0.7")
novo_df, coluna = remove_altas_correlações(titanic, 0.7)
if coluna:
    print("A coluna {} foi removida".format(coluna))
else:
    print("Nenhuma coluna removida")


# Remover atributos constantes
print("Removendo colunas constantes")
colunas_constantes = novo_df.apply(lambda x: len(x.value_counts()) == 1)
if (sum(colunas_constantes) > 0):
    print("Colunas constantes removidas:")
    for c in novo_df.columns[colunas_constantes]:
        print(c)
    novo_df = novo_df.loc[:, ~colunas_constantes]
else:
    print("Nenhuma coluna constante")


# ## 5. Limpar os dados

print("Tarefa 5: Limpar os dados")

# Preenchendoos valores de Fare acima de 500 com o Fare máximo
novo_df.loc[novo_df.Fare > 500, 'Fare'] = novo_df.Fare.max()


# Preencher os valores vazios que restam

# No caso de Embarked preenche com o valor mais comum
most_common = titanic.Embarked.value_counts().index[0]
novo_df.Embarked.fillna(value=most_common, inplace=True)


# Transformando variáveis categórias em dummy
novo_df = pd.get_dummies(novo_df, columns=['Embarked', 'Sex'])


# Para preencher os faltantes de idade faço uma regressão linear

X = novo_df.loc[novo_df.Age.notnull(), novo_df.columns.difference(['Name',
                'Ticket', 'PassengerId', 'Pclass', 'Survived'])].copy()
Y = X.Age
X.drop(['Cabin', 'Age'], axis=1, inplace=True)
```

```
    linreg = LinearRegression()
    linreg.fit(X, Y)

    X2 = novo_df.loc[novo_df.Age.isnull(), :].copy()
    X2.drop(['Cabin', 'Age', 'Name', 'Ticket', 'PassengerId',
             'Pclass', 'Survived'], axis=1, inplace=True)
    s = pd.Series(linreg.predict(X2))
    s.index = novo_df.Age[novo_df.Age.isnull()].index
    novo_df.Age.fillna(value=s, inplace=True)


    # A cabine em si é uma variável com elevada cardinalidade.
    # O melhor talvez seja o tipo de cabine representado pela letra na cabine
    def gera_CabinType(cabin):
        from numbers import Number

        if isinstance(cabin, Number):
            return 'X'
        elif ' ' in cabin:  # Mais de uma cabine
            cabins = cabin.split()  # Separa todas as cabines em uma lista
            cabins = [c[0] for c in cabins]  # Pega o 1º caractere de cada cabine
            return ''.join(set(cabins))  # Remove duplicts e junta todos CabinTypes
        else:
            return cabin[0]

    cabin_types = novo_df['Cabin'].apply(gera_CabinType)
    cabin_types.name = 'Cabin_Type'
df_final = pd.concat([novo_df.drop(['Cabin'], axis=1), cabin_types], axis=1)
```