JPNB1_Fornecedores

August 22, 2023

ANÁLISE DAS CONTRATAÇÕES NO ESTADO DE SANTA CATARINA:

JPNB 01 - PRÉ-PROCESSAMENTO DOS DADOS DE FORNECEDORES

Autor: Maurício Vasconcellos Leão Lyrio, Dr. | E-mail: mauricio@vll.adm.br | Página Oficial: www.vll.adm.br

---

1 Instalação das bibliotecas

[1]:
```
# Manipulação de dados
import pandas as pd
import numpy as np

# Visualização de dados
import matplotlib.pyplot as plt
import seaborn as sns

# Ignorar warnings
import warnings
warnings.filterwarnings('ignore')

# Versões dos pacotes utilizados neste Jupyter notebook
#!pip install -q -U watermark
%reload_ext watermark
%watermark -a "Mauricio Vasconcellos Leão Lyrio | vll.adm.br" --iversions
```

```
Author: Mauricio Vasconcellos Leão Lyrio | vll.adm.br

numpy     : 1.24.3
matplotlib: 3.4.1
```

```
pandas    : 1.5.3
seaborn   : 0.12.2
```

---

2 Carregamento dos datasets

Para iniciar nosso processo de análise de dados iremos **carregar o dataset com os dados cadastrais de fornecedores** recebido por meio de solicitação via LAI. Esse procedimento carrega o dataset do arquivo .csv original e o armazena em um dataframe pandas denominado ***df***, para que possamos manipulá-lo posteriormente.

[2]:
```
# Carregando o dataset de fornecedores
df = pd.read_csv('datasets/ELIC_fornecedores_cadastro.csv')
```

---

3 Análise exploratória dos dados

Após carregar o dataset damos início ao processo de análise exploratório, buscando **analisar a qualidade e integridade dos dados**. O processo de análise exploratório nos ajuda a ter uma visão geral do dataset e que tipo de pré-processamento precisaremos realizar nos dados a fim de deixá-los prontos para as etapas posteriores.

[3]:
```
# Verificando se o dataset foi carregado corretamente e seu tipo
type(df)
```

[3]:
```
pandas.core.frame.DataFrame
```

[4]:
```
# Verificando o formato do dataset
df.shape
```

[4]:
```
(358232, 8)
```

[5]:
```
# Listando as colunas do dataset
df.columns
```

[5]:
```
Index(['cnpj', 'razao_social', 'porte', 'cidade', 'uf', 'pais', 'situacao',
       'produtos_habilitados'],
      dtype='object')
```

[6]:
```
# Listando as informações gerais do dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 358232 entries, 0 to 358231
Data columns (total 8 columns):
 #   Column                Non-Null Count   Dtype
```

```
---  ------                 --------------   -----
 0   cnpj                   358232 non-null  int64
 1   razao_social           358232 non-null  object
 2   porte                  358232 non-null  object
 3   cidade                 358101 non-null  object
 4   uf                     358101 non-null  object
 5   pais                   358101 non-null  object
 6   situacao               358232 non-null  object
 7   produtos_habilitados   357273 non-null  object
dtypes: int64(1), object(7)
memory usage: 21.9+ MB
```

Até o momento conseguimos carregar os dados, verificar o tamanho do dataset, suas colunas, o tipo de dado de cada coluna e se existem valores ausentes. Pelo info do dataset é possível perceber que existem campos com valores nulos. Precisaremos definir o que fazer com esses campos e que tipo de tratamento iremos dar para os valores nulos, voltaremos a discutir essa questão na fase do pré-processamento de dados.

Pelo info do dataset também é possível perceber que quase todos os campos são não-numéricos. O único campo numérico é o ***cnpj***, que, na verdade, não é uma variável quantitativa e sim um código que representa o cadastro de pessoa jurídica do fornecedor. Posteriormente iremos ajustar o tipo de dado desse campo, por hora iremos somente descrever os demais campos de nosso dataset e visualizar uma amostra dos dados.

[7]:
```
# Descrevendo os dados não-numéricos
df.describe(include=object)
```

[7]:
```
                         razao_social   porte         cidade             uf  \
count                          358232  358232         358101         358101
unique                          20121       3           1181             29
top     HAKA COMERCIAL DO BRASIL LTDA  ME/EPP  FLORIANOPOLIS  Santa Catarina
freq                              851  236367          31761         173193

          pais       situacao  \
count   358101         358232
unique       3              5
top     Brasil  Empresa ativa
freq    358082         256961

                                       produtos_habilitados
count                                                357273
unique                                                  859
top     1301 - Equipamentos, programas e suprimentos d…
freq                                                   2697
```

[8]:
```
df.head()
```

[8]:
```
             cnpj                                    razao_social  \
0   6942591000100                          MCG AGUIAR CARTUCHOS ME
1   5325332000160  Global Multimídia Comércio de Eletroeletrônico…
2   9349162000104            TEXAS INFORMATICA E PRODUTOS LTDA. EPP
3  13000035000172                         TARCIANE LOHN BOECHAT EPP
4  73977480000119                       COMERCIAL STORINNY LTDA EPP

           porte                      cidade             uf    pais  \
0         ME/EPP     SAO JOSE DO RIO PRETO       São Paulo  Brasil
1         ME/EPP                  SAO PAULO       São Paulo  Brasil
2  NÃO INFORMADO                    VITORIA  Espírito Santo  Brasil
3         ME/EPP  SANTO AMARO DA IMPERATRIZ  Santa Catarina  Brasil
4         ME/EPP                 PORTO BELO  Santa Catarina  Brasil

        situacao                                produtos_habilitados
0  Empresa ativa  1303 - Equipamentos, programas e suprimentos d…
1  Empresa ativa  1303 - Equipamentos, programas e suprimentos d…
2  Empresa ativa  1303 - Equipamentos, programas e suprimentos d…
3  Empresa ativa  1303 - Equipamentos, programas e suprimentos d…
4  Empresa ativa  1303 - Equipamentos, programas e suprimentos d…
```

Com a visualização de uma amostra do dataset finalizamos a análise exploratória. Outros tipos de análise poderiam ser feitos nessa fase, porém, para nosso objetivo de preparar o dataset o que vimos até agora é suficiente. Passemos então à próxima fase do processo, o pré-processamento dos dados.

---

4 Pré-processamento

Na fase de análise exploratória identificamos que nosso dataset possui campos nulos e também que um dos campos foi definido de forma equivocada como numérico. Vamos agora tratar esses problemas e também analisar a necessidade de outros tipos de transformação de dados. Comecemos com a limpeza dos dados.

4.1 Consolidação

4.2 Limpeza

Conforme visto anteriormente, nosso dataset possui uma série de campos com valores nulos. Vamos analisar melhor essa situação e definir o que fazer com esses valores. para isso criaremos uma nova **tabela com a distribuição percentual de valores nulos por coluna**.

[9]:
```
# Criando uma tabela com a distribuição percentual dos valores nulos por coluna.
# Criando uma lista vazia para armazenar as informações de nome e tipo de␣
 ↪coluna.
colunas_info = []

# Iterando pelas colunas do dataset
```

```
for coluna in df.columns:
    coluna_nome = coluna
    coluna_tipo = df[coluna].dtype
    coluna_nulos = df[coluna].isnull().sum()
    coluna_nulos_perc = (coluna_nulos/len(df))*100
    colunas_info.
  ↪append((coluna_nome,coluna_tipo,coluna_nulos,coluna_nulos_perc))

# Criando um novo dataframe e exibindo as informações das colunas
df_colunas_info = pd.DataFrame(colunas_info, columns=['Coluna','Tipo','Q Nulo',␣
  ↪'% Nulo'])
print(df_colunas_info)
```

```
                Coluna    Tipo  Q Nulo    % Nulo
0                 cnpj   int64       0  0.000000
1         razao_social  object       0  0.000000
2                porte  object       0  0.000000
3               cidade  object     131  0.036568
4                   uf  object     131  0.036568
5                 pais  object     131  0.036568
6            situacao  object       0  0.000000
7  produtos_habilitados  object     959  0.267704
```

Com a nova tabela fica mais fácil evidenciar os valores ausentes do dataset. No caso, as colunas ***cidade***, ***uf***, ***pais*** e ***produtos_habilitados*** apresentam valores ausentes. Em projetos de data-science, em geral, utiliza-se como regra para tratamento de valores ausentes as seguintes opções:

- Para valores ausentes $>= 50\%$, descartamos a variável;
- Para valores ausentes $< 50\%$, tratar os valores ausentes;
- Para valores ausentes $< 2\%$, descartar os valores ausentes.

Apesar dessa regra geral, é importante analisar o dataframe e verificar a forma mais adequada para tratamento dos valores ausentes e, principalmente, justificar as escolhas feitas no decorrer do tratamento dos dados. Em nosso caso, **como os valores ausentes são menos de 2% dos valores dos campos vamos excluí-los**. Porém, vale salientar que, ao excluir os registros cujos campos estão ausentes perdemos parte da informação no dataset, vale refletir sobre a relevância da perda dessa informação.

Nesse caso, como os registros no dataset estão com granularidade definida em nível de produtos habilitados, acredita-se que os dados básicos dos fornecedores não serão perdidos devido à exclusão desses registros.

[10]:
```
# Criando uma cópia do dataset original e excluindo os registros com valores␣
  ↪nulos.
df1 = df.dropna()
```

[11]:
```
df1.isnull().sum()
```

[11]:
```
cnpj                    0
razao_social            0
porte                   0
cidade                  0
uf                      0
pais                    0
situacao                0
produtos_habilitados    0
dtype: int64
```

Temos agora um novo dataframe ***df1*** com os registros com valore ausentes excluídos. Mantivemos em memória o dataset original para o caso de queremos retornar à essa versão em outro momento. Passemos agora à tranformação dos dados.

4.3 Transformação

---

5 Geração de dados de saída (Data output)

5.1 Gravação em arquivos

5.2 Gravação em DB relacional

5.3 Gravação em datalake (BD não-relacional)

---

[12]:
```
# Versão da linguagem Python e arquitetura do Jupyter Notebook
import platform
print('Versão da linguagem Python utilizada neste notebook:', platform.
 ↪python_version())
print('Arquitetura do Jupyter utilizada neste notebook:', platform.
 ↪architecture()[0])
```

```
Versão da linguagem Python utilizada neste notebook: 3.8.5
Arquitetura do Jupyter utilizada neste notebook: 64bit
```