Projeto4-Aulas

November 16, 2022

1 Data Science Academy

2 Análise de Dados com Linguagem Python

2.1 Projeto 4

2.2 Análise de Dados Para Campanhas de Marketing de Instituições Financeiras

Não tenha pressa de chegar ao final. O aprendizado não está no final. O aprendizado está na jornada. Aproveite a jornada!

2.3 Pré-Requisitos

Recomendamos que você tenha concluído pelo menos os 5 primeiros capítulos do curso gratuito de Python Fundamentos Para Análise de Dados.

2.4 Instalando e Carregando os Pacotes

[1]:
```
# Versão da Linguagem Python
from platform import python_version
print('Versão da Linguagem Python Usada Neste Jupyter Notebook:',␣
 ↪python_version())
```

```
Versão da Linguagem Python Usada Neste Jupyter Notebook: 3.8.8
```

[2]:
```
# Para atualizar um pacote, execute o comando abaixo no terminal ou prompt de␣
 ↪comando:
# pip install -U nome_pacote

# Para instalar a versão exata de um pacote, execute o comando abaixo no␣
 ↪terminal ou prompt de comando:
# !pip install nome_pacote==versão_desejada

# Depois de instalar ou atualizar o pacote, reinicie o jupyter notebook.

# Instala o pacote watermark.
# Esse pacote é usado para gravar as versões de outros pacotes usados neste␣
 ↪jupyter notebook.
# !pip install -q -U watermark
```

[3]:
```
# Imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

[4]:
```
# Versões dos pacotes usados neste jupyter notebook
%reload_ext watermark
%watermark -a "Data Science Academy" --iversions
```

```
Author: Data Science Academy

seaborn   : 0.11.1
matplotlib: 3.4.3
numpy     : 1.21.2
pandas    : 1.3.3
```

2.5 Carregando os Dados

[5]:
```
# Carrega o dataset
df = pd.read_csv("dados/dataset.csv")
```

[6]:
```
# Shape
df.shape
```

[6]:
```
(45211, 19)
```

[7]:
```
# Amostra
df.head()
```

[7]:

	customerid	age	salary	balance	marital	jobedu	targeted	default	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	response
0	1	58.0	100000.0	2143	married	management,tertiary	yes	no	yes	no	unknown	5	may, 2017	261 sec	1	-1	0	unknown	no
1	2	44.0	60000.0	29	single	technician,secondary	yes	no	yes	no	unknown	5	may, 2017	151 sec	1	-1	0	unknown	no
2	3	33.0	120000.0	2	married	entrepreneur,secondary	yes	no	yes	yes	unknown	5	may, 2017	76 sec	1	-1	0	unknown	no
3	4	47.0	20000.0	1506	married	blue-collar,unknown	no	no	yes	no	unknown	5	may, 2017	92 sec	1	-1	0	unknown	no
4	5	33.0	0.0	1	single	unknown,unknown	no	no	no	no	unknown	5	may, 2017	198 sec	1	-1	0	unknown	no

2.6 Análise Exploratória

[8]:
```
# Info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 19 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   customerid  45211 non-null  int64
 1   age         45191 non-null  float64
 2   salary      45185 non-null  float64
 3   balance     45211 non-null  int64
```

```
 4   marital     45211 non-null  object
 5   jobedu      45211 non-null  object
 6   targeted    45211 non-null  object
 7   default     45211 non-null  object
 8   housing     45211 non-null  object
 9   loan        45211 non-null  object
 10  contact     45211 non-null  object
 11  day         45211 non-null  int64
 12  month       45161 non-null  object
 13  duration    45211 non-null  object
 14  campaign    45211 non-null  int64
 15  pdays       45211 non-null  int64
 16  previous    45211 non-null  int64
 17  poutcome    45211 non-null  object
 18  response    45181 non-null  object
dtypes: float64(2), int64(6), object(11)
memory usage: 6.6+ MB
```

[9]:
```
# Temos valores nulos? Sim ou não?
df.isna().any()
```

[9]:
```
customerid    False
age            True
salary         True
balance       False
marital       False
jobedu        False
targeted      False
default       False
housing       False
loan          False
contact       False
day           False
month          True
duration      False
campaign      False
pdays         False
previous      False
poutcome      False
response       True
dtype: bool
```

[10]:
```
# Temos valores nulos? Quantos?
df.isna().sum()
```

[10]:
```
customerid     0
age           20
salary        26
```

```
balance        0
marital        0
jobedu         0
targeted       0
default        0
housing        0
loan           0
contact        0
day            0
month         50
duration       0
campaign       0
pdays          0
previous       0
poutcome       0
response      30
dtype: int64
```

[11]:
```
# Não usaremos a coluna ID. Vamos removê-la.
df.drop(["customerid"], axis = 1, inplace = True)
```

[12]:
```
# Colunas
df.columns
```

[12]:
```
Index(['age', 'salary', 'balance', 'marital', 'jobedu', 'targeted', 'default',
       'housing', 'loan', 'contact', 'day', 'month', 'duration', 'campaign',
       'pdays', 'previous', 'poutcome', 'response'],
      dtype='object')
```

Exercício 1: A coluna "jobedu" parece ter duas informações. Vamos separar em duas colunas.

[13]:
```
# Coloque sua solução aqui
```

2.7 Tratamento de Valores Ausentes

Vamos primeiro tratar a variável que representa a idade.

[14]:
```
# Valores ausentes no dataframe
df.isna().any()
```

[14]:
```
age           True
salary        True
balance      False
marital      False
jobedu       False
targeted     False
default      False
```

```
housing     False
loan        False
contact     False
day         False
month        True
duration    False
campaign    False
pdays       False
previous    False
poutcome    False
response     True
dtype: bool
```

[15]:
```
# Valores ausentes da variável age
df.age.isnull().sum()
```

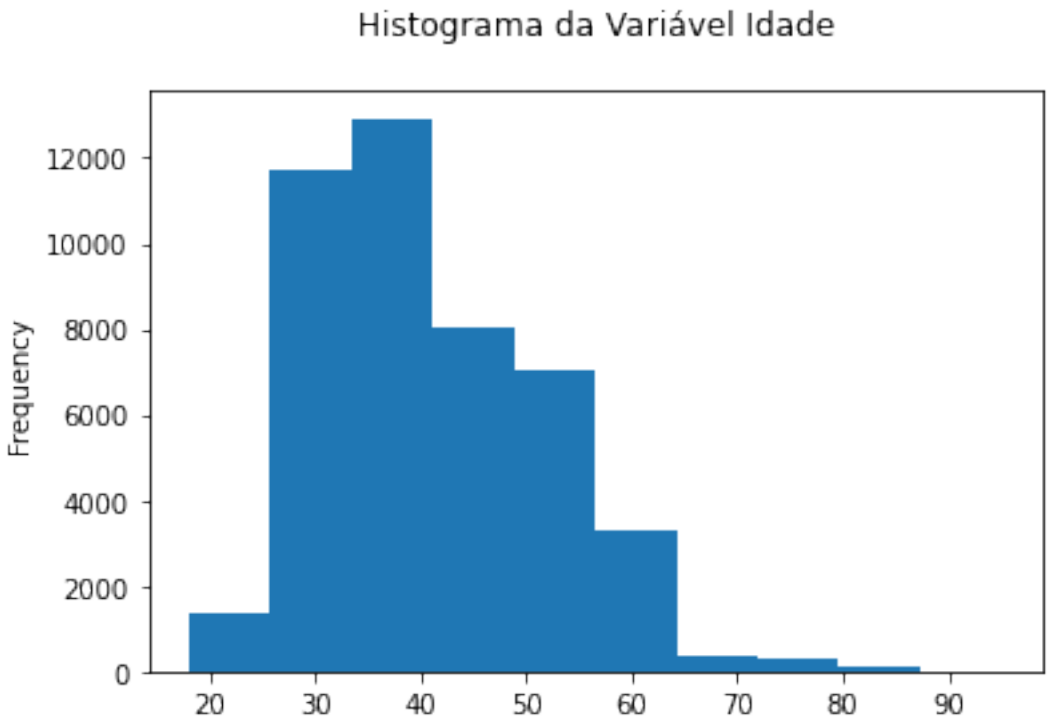
[15]:
```
20
```

[16]:
```
# Calcula o percentual de valores ausentes na variável age
df.age.isnull().mean()*100
```

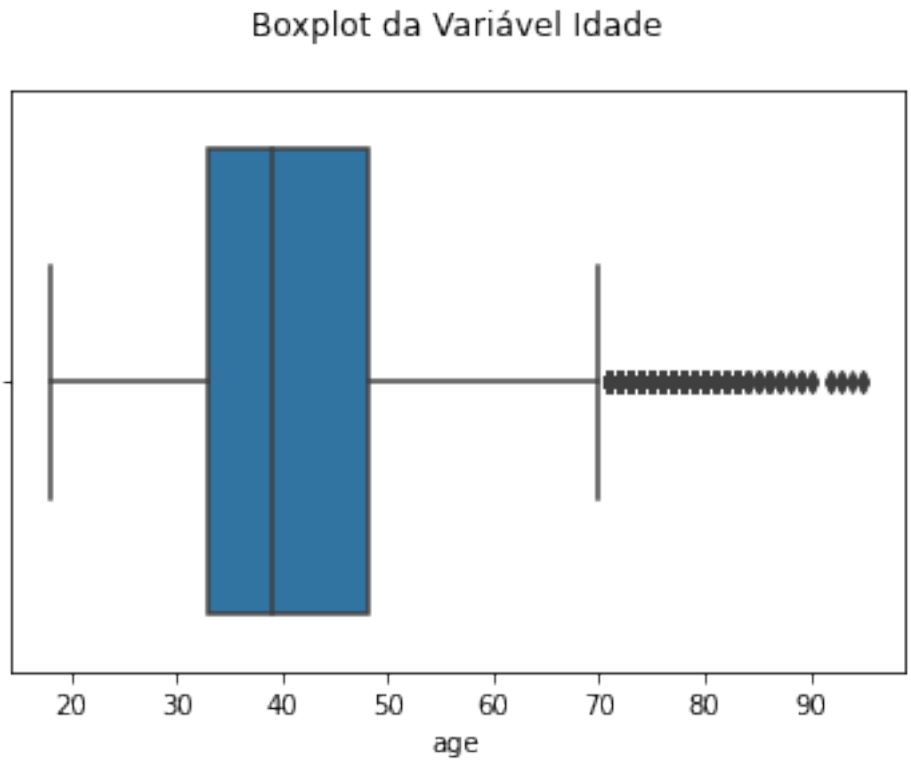
[16]:
```
0.0442370219636814
```

Como o percentual é baixo não podemos eliminar a coluna. Podemos então eliminar os registros com valores ausentes (nesse caso perderíamos 20 linhas no dataset) ou podemos aplicar imputação. Vamos usar a segunda opção.

[17]:
```
# Histograma
df.age.plot(kind = "hist")
plt.title("Histograma da Variável Idade\n")
plt.show()
```



[18]:
```
# Boxplot
sns.boxplot(df.age)
plt.title("Boxplot da Variável Idade\n")
plt.show()
```



[19]:
```
# Vamos verificar qual é a média de idade.
df.age.mean()
```

[19]:
```
40.93565090394105
```

[20]:
```
# Vamos verificar qual é a mediana, valor do meio da distribuição quando os␣
↪dados estão ordenados.
df.age.median()
```

[20]:
```
39.0
```

[21]:
```
# Vamos verificar qual é a moda, o valor que aparece com mais frequência.
df.age.mode()
```

[21]:
```
0    32.0
dtype: float64
```

Exercício 2: Vamos imputar os valores ausentes da variável age com uma medida de tendência central. Escolha uma das medidas, aplique a imputação e justifique sua escolha. Deixamos a variável como float ou como int? Se convertemos, fazemos isso antes ou depois da imputação?

[22]:
```
# Coloque sua solução aqui
```

2.8 Tratamento de Valores Ausentes

Vamos agora tratar a variável que representa o mês.

[23]:
```
# Valores ausentes na variável
df.month.isnull().sum()
```

[23]:
```
50
```

[24]:
```
# Percentual de valores ausentes
df.month.isnull().mean()*100
```

[24]:
```
0.11059255490920351
```

Como o percentual é menor que 30% não podemos eliminar a coluna. Podemos então eliminar os registros com valores ausentes (nesse caso perderíamos 50 linhas no dataset) ou podemos aplicar imputação. Vamos usar a segunda opção.

[25]:
```
# Tipo da variável
df.month.dtypes
```

[25]:
```
dtype('O')
```

[26]:
```
# Categorias da variável
df.month.value_counts()
```

[26]:
```
may, 2017    13747
jul, 2017     6888
aug, 2017     6240
jun, 2017     5335
nov, 2017     3968
apr, 2017     2931
feb, 2017     2646
jan, 2017     1402
oct, 2017      738
sep, 2017      576
mar, 2017      476
dec, 2017      214
Name: month, dtype: int64
```

Exercício 3: Vamos imputar os valores ausentes da variável month. Escolha uma estratégia e aplique no dataset.

[27]:
```
# Coloque sua solução aqui
```

2.9 Tratamento de Valores Ausentes

Vamos agora tratar a variável que representa o salário.

[28]:
```
# Valores ausentes na variável
df.salary.isnull().sum()
```

[28]:
```
26
```

[29]:
```
# Calcula o percentual de valores ausentes na variável salary
df.salary.isnull().mean()*100
```

[29]:
```
0.05750812855278583
```

Como o percentual é baixo não podemos eliminar a coluna. Podemos então eliminar os registros com valores ausentes (nesse caso perderíamos 26 linhas no dataset) ou podemos aplicar imputação. Vamos usar a segunda opção.

Mas espere. Vamos checar algo aqui.

[30]:
```
df.head()
```

[30]:

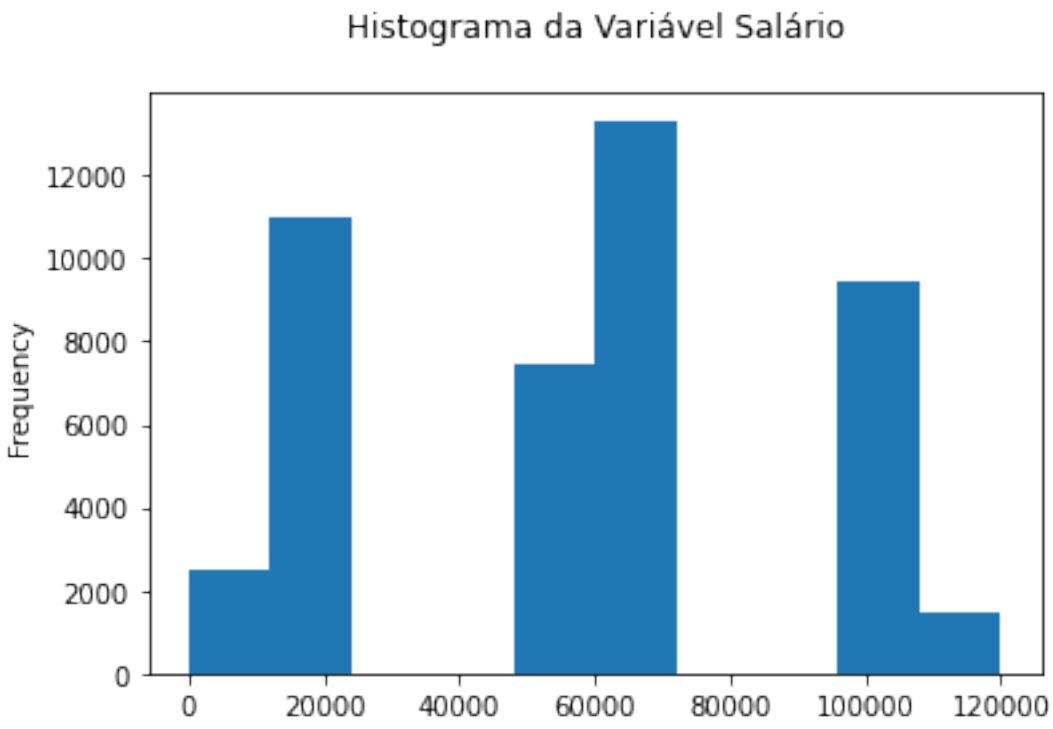
	age	salary	balance	marital	jobedu	targeted	default	\
0	58.0	100000.0	2143	married	management,tertiary	yes	no	
1	44.0	60000.0	29	single	technician,secondary	yes	no	
2	33.0	120000.0	2	married	entrepreneur,secondary	yes	no	
3	47.0	20000.0	1506	married	blue-collar,unknown	no	no	
4	33.0	0.0	1	single	unknown,unknown	no	no	

	housing	loan	contact	day	month	duration	campaign	pdays	previous	\
0	yes	no	unknown	5	may, 2017	261 sec	1	-1	0	
1	yes	no	unknown	5	may, 2017	151 sec	1	-1	0	
2	yes	yes	unknown	5	may, 2017	76 sec	1	-1	0	
3	yes	no	unknown	5	may, 2017	92 sec	1	-1	0	
4	no	no	unknown	5	may, 2017	198 sec	1	-1	0	

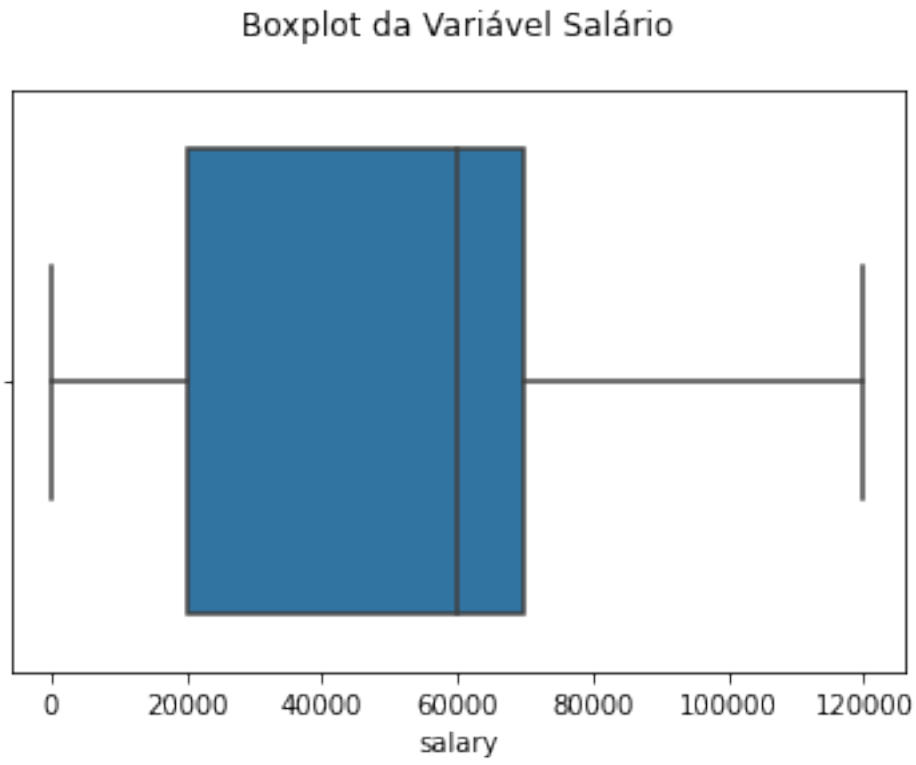
	poutcome	response
0	unknown	no
1	unknown	no
2	unknown	no
3	unknown	no
4	unknown	no

Existe salário igual a zero? Não. O valor zero é provavelmente um valor ausente (confirmar com a área de negócio).

[31]:
```
# Histograma
df.salary.plot(kind = "hist")
plt.title("Histograma da Variável Salário\n")
plt.show()
```



[32]:
```
# Boxplot
sns.boxplot(df.salary)
plt.title("Boxplot da Variável Salário\n")
plt.show()
```



[33]:
```
# Vamos verificar qual é a média de idade.
df.salary.mean()
```

[33]:
```
57008.65331415293
```

[34]:
```
# Vamos verificar qual é a mediana.
df.salary.median()
```

[34]:
```
60000.0
```

[35]:
```
# Vamos verificar qual é a moda.
df.salary.mode()
```

[35]:
```
0    20000.0
dtype: float64
```

Exercício 4: Vamos imputar os valores ausentes da variável salary com uma medida de tendência central. Precisamos também tratar os valores iguais a zero. Escolha sua estratégia, aplique a imputação e justifique sua escolha.

[36]:
```
# Coloque sua solução aqui
```

2.10 Tratamento de Valores Ausentes

Vamos agora tratar a variável que representa a resposta (variável alvo).

[37]:
```
df.head()
```

[37]:
```
    age    salary  balance  marital                jobedu targeted default  \
0  58.0  100000.0     2143  married     management,tertiary      yes      no
1  44.0   60000.0       29   single    technician,secondary      yes      no
2  33.0  120000.0        2  married  entrepreneur,secondary      yes      no
3  47.0   20000.0     1506  married     blue-collar,unknown       no      no
4  33.0       0.0        1   single         unknown,unknown       no      no

  housing loan  contact  day      month duration  campaign  pdays  previous  \
0     yes   no  unknown    5  may, 2017  261 sec         1     -1         0
1     yes   no  unknown    5  may, 2017  151 sec         1     -1         0
2     yes  yes  unknown    5  may, 2017   76 sec         1     -1         0
3     yes   no  unknown    5  may, 2017   92 sec         1     -1         0
4      no   no  unknown    5  may, 2017  198 sec         1     -1         0

  poutcome response
0  unknown       no
1  unknown       no
2  unknown       no
3  unknown       no
4  unknown       no
```

[38]:
```
# Valores ausentes
df.response.isnull().sum()
```

[38]:
```
30
```

[39]:
```
# Calcula o percentual
df.response.isnull().mean()*100
```

[39]:
```
0.0663555329455221
```

Como o percentual é baixo (e a variável é o alvo da nossa análise) não podemos eliminar a coluna. Podemos então eliminar os registros com valores ausentes (nesse caso perderíamos 30 linhas no dataset) ou podemos aplicar imputação.

Exercício 5: Escolha sua estratégia, aplique e justifique sua escolha.

[40]:
```
# Coloque sua solução aqui
```

2.11 Tratamento de Valores Ausentes

Vamos agora tratar a variável pdays.

[41]:
```
# Valores ausentes
df.pdays.isnull().sum()
```

[41]:
```
0
```

[42]:
```
# Describe
df.pdays.describe()
```

[42]:
```
count    45211.000000
mean        40.197828
std        100.128746
min         -1.000000
25%         -1.000000
50%         -1.000000
75%         -1.000000
max        871.000000
Name: pdays, dtype: float64
```

-1 indica valor ausente

[43]:
```
# Vamos fazer relace de -1 por NaN
df.pdays = df.pdays.replace({-1.0:np.NaN})
```

[44]:
```
# Valores ausentes
df.pdays.isnull().sum()
```

[44]:
```
36954
```

[45]:
```
# Calcula o percentual
df.pdays.isnull().mean()*100
```

[45]:
```
81.73674548229414
```

Exercício 6: Escolha sua estratégia, aplique e justifique sua escolha.

[46]:
```
# Coloque sua solução aqui
```

3 Fim