

# **Desafio**

## **Previsão de venda para itens da Renner SA**

**Lucas Gomes**

Student in Data Science and AI

**Luciano Farias**

Student in Data Science and AI

Maio, 2023

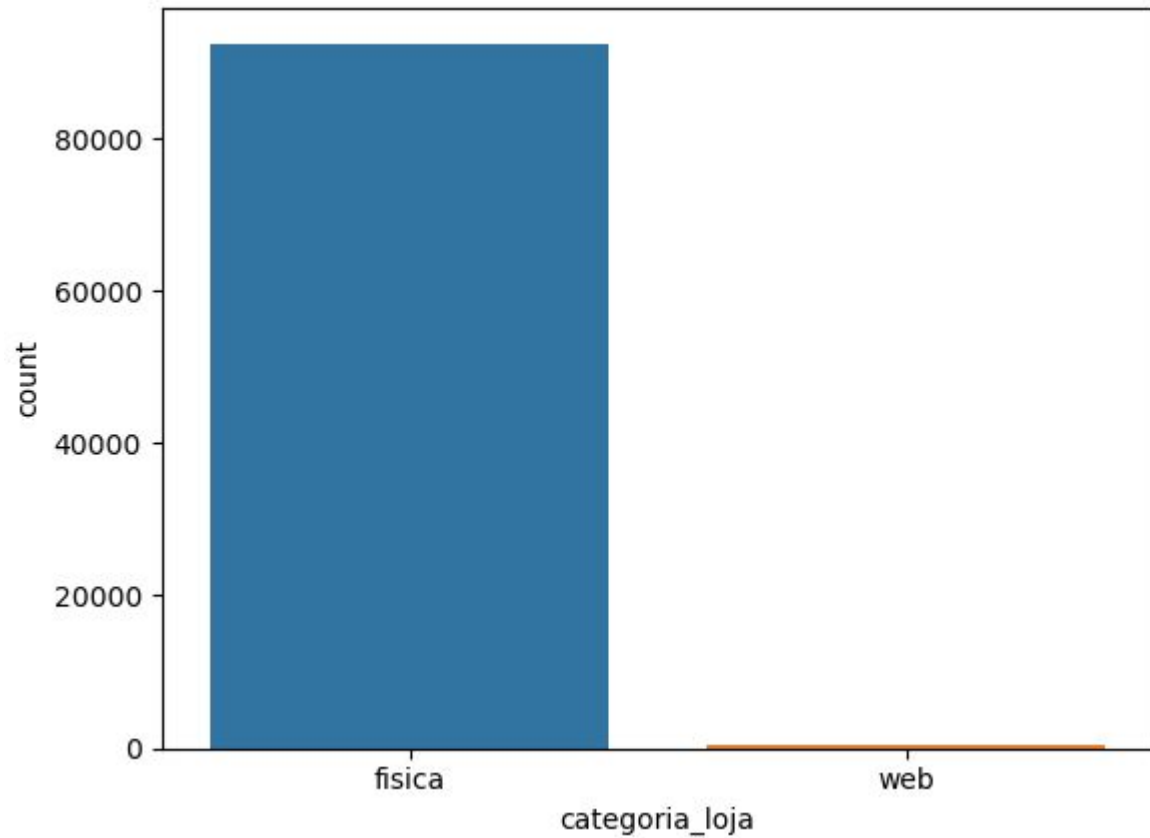
Porto Alegre, Brasil

# Tópicos:

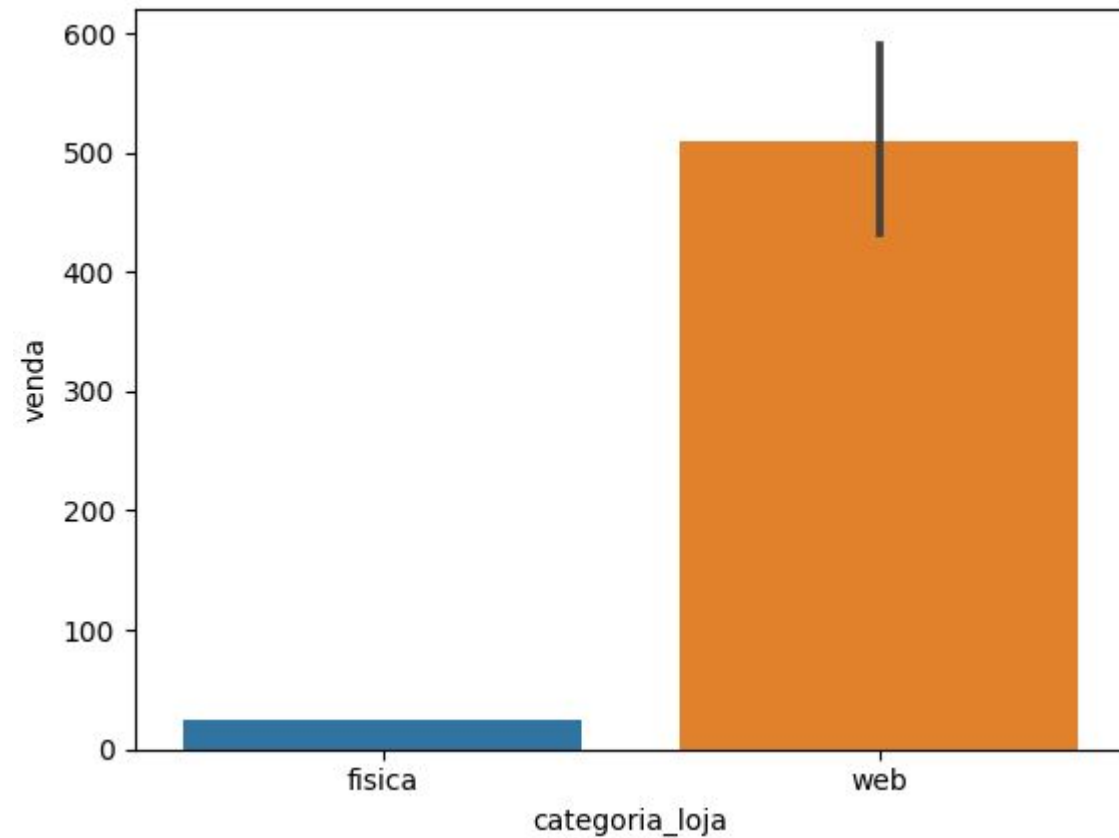
- Tecnologias usadas/Extração dos dados;
- Exploração dos dados;
- Tratamentos iniciais;
- Primeiros insights

- Python;
- Pandas;
- Matplotlib;
- Seaborn.

Quantidade de lojas físicas/web



Quantidade de vendas em lojas físicas/web



## Quantidades de vendas negativas

```
df[df['venda'] < 0]
```

	item	cod_loja	semana_comercial	data_semana_comercial	venda	categoria_loja	localidade	loja_tamanho	clima
	900	100342	12	201743	2017-10-22	-1	fisica	br	G SUL
	1596	100342	321	201943	2019-10-27	-2	fisica	br	P SPO
	1791	100342	270	202007	2020-02-16	-1	fisica	br	M SUL
	1837	100342	273	202138	2021-09-19	-2	fisica	br	P RIO
	1983	100342	111	202203	2022-01-16	-1	fisica	br	M RIO
	...	...	...	...	...	...	...	...	...
	92827	100342	58	202001	2020-01-05	-1	fisica	br	G RIO
	92846	100342	323	202145	2021-11-07	-1	fisica	br	P NOR
	92850	100342	171	201826	2018-07-01	-1	fisica	br	P SPO
	92858	100342	312	201745	2017-11-05	-1	fisica	br	M SUL
	92875	100342	163	201950	2019-12-15	-1	fisica	br	M SPO

520 rows × 9 columns

## Colunas desnecessárias

	item	cod_loja	semana_comercial	data_semana_comercial	venda	categoria_loja	localidade	loja_tamanho	clima
0	100342	58	202130	2021-07-25	51	fisica	br	G	RIO
1	100342	63	202221	2022-05-22	86	fisica	br	G	RIO
2	100342	84	202131	2021-08-01	62	fisica	br	G	RIO
3	100342	2	202222	2022-05-29	171	fisica	br	G	SUL
4	100342	407	202223	2022-06-05	188	web	br	GG	W

Linhas com cod\_loja e data\_semana\_comercial iguais, mas com quantidades de venda diferentes



The screenshot shows a database query builder interface. The top tab is 'Planilha' and the bottom tab is 'Query Builder'. The SQL query is displayed in a text area:

```
select * from renner20231 where cod_loja =1  
and extract(month from data_semana_comercial) = 12  
and extract(day from data_semana_comercial) > 24  
order by data_semana_comercial, semana_comercial;
```

Below the query, the 'Resultado da Consulta' tab is active, showing the results of the query. The status bar indicates 'Todas as Linhas Extraídas: 8 em 0,018 segundos'. The results are displayed in a table with 10 columns: ITEM, COD\_LOJA, SEMANA\_COMERCIAL, DATA\_SEMANA\_COMERCIAL, VENDA, CATEGORIA\_LOJA, LOCALIDADE, LOJA\_TAMANHO, and CLIMA.

	ITEM	COD_LOJA	SEMANA_COMERCIAL	DATA_SEMANA_COMERCIAL	VENDA	CATEGORIA_LOJA	LOCALIDADE	LOJA_TAMANHO	CLIMA
1	100342	1	201853	2017-12-31	19	fisica	br	G	SUL
2	100342	1	201852	2018-12-30	2	fisica	br	G	SUL
3	100342	1	201952	2018-12-30	6	fisica	br	G	SUL
4	100342	1	201952	2019-12-29	2	fisica	br	G	SUL
5	100342	1	202052	2019-12-29	1	fisica	br	G	SUL
6	100342	1	202052	2020-12-27	2	fisica	br	G	SUL
7	100342	1	202152	2020-12-27	3	fisica	br	G	SUL
8	100342	1	202152	2021-12-26	1	fisica	br	G	SUL

## Drop das colunas desnecessárias

```
del df['item']  
del df['semana_comercial']  
df.head()
```

✓ 0.0s

	cod_loja	data_semana_comercial	venda	categoria_loja	localidade	loja_tamanho	clima
0	58	2021-07-25	51	fisica	br	G	RIO
1	63	2022-05-22	86	fisica	br	G	RIO
2	84	2021-08-01	62	fisica	br	G	RIO
3	2	2022-05-29	171	fisica	br	G	SUL
4	407	2022-06-05	188	web	br	GG	W



Quantidades negativas para zero

```
df['venda'] = df[df['venda'] < 0] = 0  
df[df['venda'] < 0]
```

cod_loja	data_semana_comercial	venda	categoria_loja	localidade	loja_tamanho	clima
----------	-----------------------	-------	----------------	------------	--------------	-------

[+ Code](#)[+ Markdown](#)

Somando as quantidades de venda com cod\_loja e data\_sema\_comercial iguais

```
# Agrupa o dataset por data_semana_comercial, mas somando somente as vendas e mantendo as outras colunas iguais
df_agrupado = df.groupby('data_semana_comercial').agg(
    {'venda': 'sum',
     'cod_loja': 'first',
     'categoria_loja': 'first',
     'localidade': 'first',
     'loja_tamanho': 'first',
     'clima': 'first'}).reset_index()
```

✓ 0.0s

Alterando as variáveis categóricas para numéricas

```
df_agrupado.iloc[:, 4] = localidade_encoder.fit_transform(df_agrupado.iloc[:, 4])  
df_agrupado.iloc[:, 5] = loja_tamanho_encoder.fit_transform(df_agrupado.iloc[:, 5])  
df_agrupado.iloc[:, 6] = clima_encoder.fit_transform(df_agrupado.iloc[:, 6])
```

✓ 0.0s

```
df_agrupado.head()
```

✓ 0.0s

	data_semana_comercial	venda	cod_loja	categoria_loja	localidade	loja_tamanho	clima
0	2017-01-01	0	47	fisica	1	2	3
1	2017-01-08	0	150	fisica	1	0	2
2	2017-01-15	0	132	fisica	1	3	4
3	2017-01-22	0	43	fisica	1	2	3
4	2017-01-29	0	67	fisica	1	2	4

Separando entre dataset de treino para loja web e fisica

```
#Separando entre dataset web e dataset fisico  
df_web = df_agrupado[df_agrupado['categoria_loja'] == 'web']  
df_fisica = df_agrupado[df_agrupado['categoria_loja'] == 'fisica']
```

✓ 0.0s

```
df_web.head(2)
```

✓ 0.0s

	data_semana_comercial	venda	cod_loja	categoria_loja	localidade	loja_tamanho	clima
36	2017-09-10	0	406	web	1	1	6
137	2019-08-18	0	406	web	1	1	6

```
df_fisica.head(2)
```

✓ 0.0s

	data_semana_comercial	venda	cod_loja	categoria_loja	localidade	loja_tamanho	clima
0	2017-01-01	0	47	fisica	1	2	3
1	2017-01-08	0	150	fisica	1	0	2

Dropando novas colunas desnecessárias em cada dataset

```
del df_web['categoria_loja']  
del df_web['clima']  
del df_web['localidade']  
df_web.head()
```

✓ 0.0s

	data_semana_comercial	venda	cod_loja	loja_tamanho
36	2017-09-10	0	406	1
137	2019-08-18	0	406	1
168	2020-03-22	0	406	1
169	2020-03-29	0	406	1
170	2020-04-05	0	406	1

```
del df_fisica['categoria_loja']  
df_fisica.head()
```

✓ 0.0s

	data_semana_comercial	venda	cod_loja	localidade	loja_tamanho	clima
0	2017-01-01	0	47	1	2	3
1	2017-01-08	0	150	1	0	2
2	2017-01-15	0	132	1	3	4
3	2017-01-22	0	43	1	2	3
4	2017-01-29	0	67	1	2	4

# PUCRS

**Obrigado**

Porto Alegre, Brasil – 2023