Previsão de faturamento Tiller Primeiro foi importada a tabela do Bigquery

In [2]:

```python
import pandas as pd

from google.cloud import bigquery
from google.oauth2 import service_account

credentials = service_account.Credentials.from_service_account_file(
    'C:/Users/ddior/Downloads/winged-helper-384014-161c1ccaa8f2.json',
    scopes=['https://www.googleapis.com/auth/cloud-platform'],
)

query_string = 'SELECT * FROM `winged-helper-384014.proj_tiller.order_line_store`;'
df = pd.read_gbq(query_string, credentials=credentials)
df.head()
```

Out[2]:

	ordem_do_pedido	pedido	data_pagamento	quantidade	preco_unitario	preco_unitario_sem_taxa	preco_total_sem_taxa	preco_total_com_taxa	taxa	desconto	categoria	no
0	74420791	15006629	2017-10-20 11:04:00+00:00	1.0	0.0	0.0	0.0	0.0	0.1	0.0	BREIZH&#39;N&#39;ROLL	N
1	157320867	34838322	2018-07-11 10:43:57+00:00	1.0	0.0	0.0	0.0	0.0	0.1	0.0	BUCKWHEAT WRAP	N
2	26665476	5636040	2017-02-10 18:00:54+00:00	1.0	0.0	0.0	0.0	0.0	0.1	0.0	cooking	B
3	193315744	43538157	2018-09-29 17:42:51+00:00	1.0	0.0	0.0	0.0	0.0	0.1	0.0	SCENT	BLACKBER
4	212658903	48482987	2018-11-10 13:12:46+00:00	1.0	0.0	0.0	0.0	0.0	0.1	0.0	BRUNCH	E

In [3]:

```python
df['data_pagamento'] = df['data_pagamento'].dt.date
df
```

Out[3]:

	ordem_do_pedido	pedido	data_pagamento	quantidade	preco_unitario	preco_unitario_sem_taxa	preco_total_sem_taxa	preco_total_com_taxa	taxa	desconto	categoria	
0	74420791	15006629	2017-10-20	1.0	0.0	0.000000	0.000000	0.0	0.100	0.0	BREIZH&#39;N&#39;ROLL	
1	157320867	34838322	2018-07-11	1.0	0.0	0.000000	0.000000	0.0	0.100	0.0	BUCKWHEAT WRAP	
2	26665476	5636040	2017-02-10	1.0	0.0	0.000000	0.000000	0.0	0.100	0.0	cooking	
3	193315744	43538157	2018-09-29	1.0	0.0	0.000000	0.000000	0.0	0.100	0.0	SCENT	BL
4	212658903	48482987	2018-11-10	1.0	0.0	0.000000	0.000000	0.0	0.100	0.0	BRUNCH	
...	...	...	...	...	...	...	...	...	...	...	...	
3924743	295057205	69272754	2019-04-21	1.0	19.0	18.009479	18.009479	19.0	0.055	0.0	Coffee beans	
3924744	315586286	74335158	2019-05-26	1.0	19.0	18.009479	18.009479	19.0	0.055	0.0	Coffee beans	
3924745	379201250	89417092	2019-08-28	1.0	19.0	18.009479	18.009479	19.0	0.055	0.0	Coffee beans	
3924746	567912049	124897239	2020-05-06	1.0	19.0	18.009479	18.009479	19.0	0.055	0.0	Coffee Sachet	
3924747	332013743	78372603	2019-06-20	1.0	19.0	18.009479	18.009479	19.0	0.055	0.0	Coffee beans	

3924748 rows × 13 columns

In [5]:

```python
df_grouped = df.groupby(['pedido', 'data_pagamento', 'id_store']).agg({
    'quantidade': 'sum',
    'preco_total_sem_taxa': 'sum',
    'preco_total_com_taxa': 'sum',
    'desconto': 'sum',
})
```

In [19]:

```python
df_clean = df_grouped.drop_duplicates(subset='pedido').reset_index(drop=True)
df_clean = df_clean[df_clean['preco_total_com_taxa'] != 0]
df_clean
```

Out[19]:

	index	pedido	data_pagamento	id_store	quantidade	preco_total_sem_taxa	preco_total_com_taxa	desconto
0	0	218270	2015-10-22	351	25.0	321.069367	362.6	0.0
1	1	218279	2015-10-22	351	3.0	19.927014	21.1	0.0
2	2	218366	2015-10-22	351	6.0	45.000000	52.0	0.0
3	3	218423	2015-10-22	351	8.0	59.090909	68.0	0.0
4	4	218549	2015-10-22	351	24.0	265.205766	295.1	0.0
...	...	...	...	...	...	...	...	...
1279485	1280460	150531724	2020-11-18	6830	4.0	40.181818	44.2	0.0
1279486	1280461	150531741	2020-11-18	6830	3.0	20.363636	22.4	0.0
1279487	1280462	150531767	2020-11-18	6830	2.0	29.090909	32.0	0.0
1279488	1280463	150531773	2020-11-18	6830	4.0	20.654545	22.7	0.0
1279489	1280464	150531828	2020-11-18	6830	8.0	54.818182	60.3	0.0

1268182 rows × 8 columns

In [23]:

```python
df_clean['data_pagamento'] = pd.to_datetime(df_clean['data_pagamento'])

df_clean['mes'] = df_clean['data_pagamento'].dt.month
df_clean['ano'] = df_clean['data_pagamento'].dt.year
df_clean
```

Out[23]:

	index	pedido	data_pagamento	id_store	quantidade	preco_total_sem_taxa	preco_total_com_taxa	desconto	mes	ano
0	0	218270	2015-10-22	351	25.0	321.069367	362.6	0.0	10	2015
1	1	218279	2015-10-22	351	3.0	19.927014	21.1	0.0	10	2015
2	2	218366	2015-10-22	351	6.0	45.000000	52.0	0.0	10	2015
3	3	218423	2015-10-22	351	8.0	59.090909	68.0	0.0	10	2015
4	4	218549	2015-10-22	351	24.0	265.205766	295.1	0.0	10	2015
...	...	...	...	...	...	...	...	...	...	...
1279485	1280460	150531724	2020-11-18	6830	4.0	40.181818	44.2	0.0	11	2020
1279486	1280461	150531741	2020-11-18	6830	3.0	20.363636	22.4	0.0	11	2020
1279487	1280462	150531767	2020-11-18	6830	2.0	29.090909	32.0	0.0	11	2020
1279488	1280463	150531773	2020-11-18	6830	4.0	20.654545	22.7	0.0	11	2020
1279489	1280464	150531828	2020-11-18	6830	8.0	54.818182	60.3	0.0	11	2020

1268182 rows × 10 columns

In [25]:

```python
from sklearn.linear_model import LinearRegression

model = LinearRegression()

X = df_clean[['mes','quantidade', 'desconto', 'id_store']]
y = df_clean['preco_total_com_taxa']

X_train = X.head(len(X)-253640)
X_test = X.tail(253640)
y_train = y.head(len(X)-253640)
y_test = y.tail(253640)

model.fit(X_train, y_train)
```

Out[25]:

```
LinearRegression()
```

In [27]:

```python
prediction = model.predict(X_test)
prediction
```

Out[27]:

```
array([ 2.20694744, 27.21216693, 14.67981479, ...,  8.55466506,
       21.0870172 , 46.1517215 ])
```

In [29]:

```python
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np

mse = np.mean((prediction - y_test) ** 2)
mae = np.mean(np.abs(prediction - y_test))
r2 = 1 - np.sum((y_test - prediction) ** 2) / np.sum((prediction - np.mean(prediction)) ** 2)

print("Mean Squared Error (MSE):", mse)
print("Mean Absolute Error (MAE):", mae)
print("R-squared (R2):", r2)
```

```
Mean Squared Error (MSE): 1093.6915081240024
Mean Absolute Error (MAE): 10.525516241731449
R-squared (R2): -1.414478743960002
```

In [33]:

```python
df_new = df_new = pd.DataFrame({'date': df_clean.tail(253640)['data_pagamento'],'loja': df_clean.tail(253640)['id_store'],'previsao_faturamento': prediction})
df_new['preco_com_taxa'] = y_test
df_new = df_new.set_index('date')
df_new
```

Out[33]:

	loja	previsao_faturamento	preco_com_taxa
date			
2019-11-11	5281	2.206947	8.0
2019-11-11	4151	27.212167	12.5
2019-11-11	4151	14.679815	4.5
2019-11-11	4151	33.478343	20.5
2019-11-11	4151	27.212167	13.5
...	...	...	...
2020-11-18	6830	21.087017	44.2
2020-11-18	6830	14.820841	22.4
2020-11-18	6830	8.554665	32.0
2020-11-18	6830	21.087017	22.7
2020-11-18	6830	46.151721	60.3

253640 rows × 3 columns

In [34]:

```python
import pandas_gbq

project_id = 'winged-helper-384014'
table_name = 'proj_tiller.previsao_de_faturamento_loja'

pandas_gbq.to_gbq(df_new, table_name, project_id=project_id, if_exists='replace')
```

```
100%|██████████| 1/1 [00:00<?, ?it/s]
```