RECOMENDAÇÃO DE RESTAURANTE

Abdul Malik de Barros
Ana Beatriz Oliveira de Macedo
Ana Carolina Zhang
Bruna Bellini Faria
Heloisa Mariani Rodrigues
Marina Lara

CDIA - MA5

CONTEXTUALIZAÇÃO E OBJETIVO DO TRABALHO

Este trabalho tem como objetivo **fornecer ao usuário uma ferramenta que permita a busca por uma lista de estabelecimentos, oferecendo informações importantes e relevantes** de acessibilidade, horário de funcionamento e localização, com a vantagem de apresentar um resumo das principais avaliações para que o usuário possa avaliar mais facilmente se o estabelecimento atende às suas necessidades.

Além disso, este projeto visa **sugerir estabelecimentos similares com aqueles que o usuário está pesquisando**, mas que possam ser de interesse do cliente, mesmo que não façam parte daquela categoria específica.

Para alcançar esses objetivos, foram utilizadas as **API's do Google Maps Reviews e Google Maps Place Results**, que fornecem dados sobre os estabelecimentos, como região, bairro, horário de funcionamento, dias de funcionamento, categoria, preço, entre outros.

Com base nas informações fornecidas, o usuário também poderá utilizar esta ferramenta para **realizar pesquisas de mercado**, como por exemplo, ao considerar a abertura de um novo estabelecimento. Ao observar as categorias e localizações de outros estabelecimentos, é possível evitar lugares de muita concorrência e explorar regiões com alta demanda e pouca concorrência.

Neste contexto, o uso da tecnologia de Processamento de Linguagem Natural (PLN) se faz essencial para a compilação e organização das informações disponíveis de forma eficiente e intuitiva para o usuário. Com isso, o presente trabalho tem o intuito de oferecer uma solução prática e inovadora para os usuários que buscam informações sobre estabelecimentos comerciais.

Neste projeto, é evidente a importância de contextualizar o objetivo do trabalho e como ele pode trazer benefícios para o usuário, como a praticidade e facilidade na busca por informações relevantes de estabelecimentos comerciais. Por fim, a utilização de modelos de análise de sentimento e de clusterização complementam o projeto, proporcionando ao usuário uma ferramenta completa e inovadora.

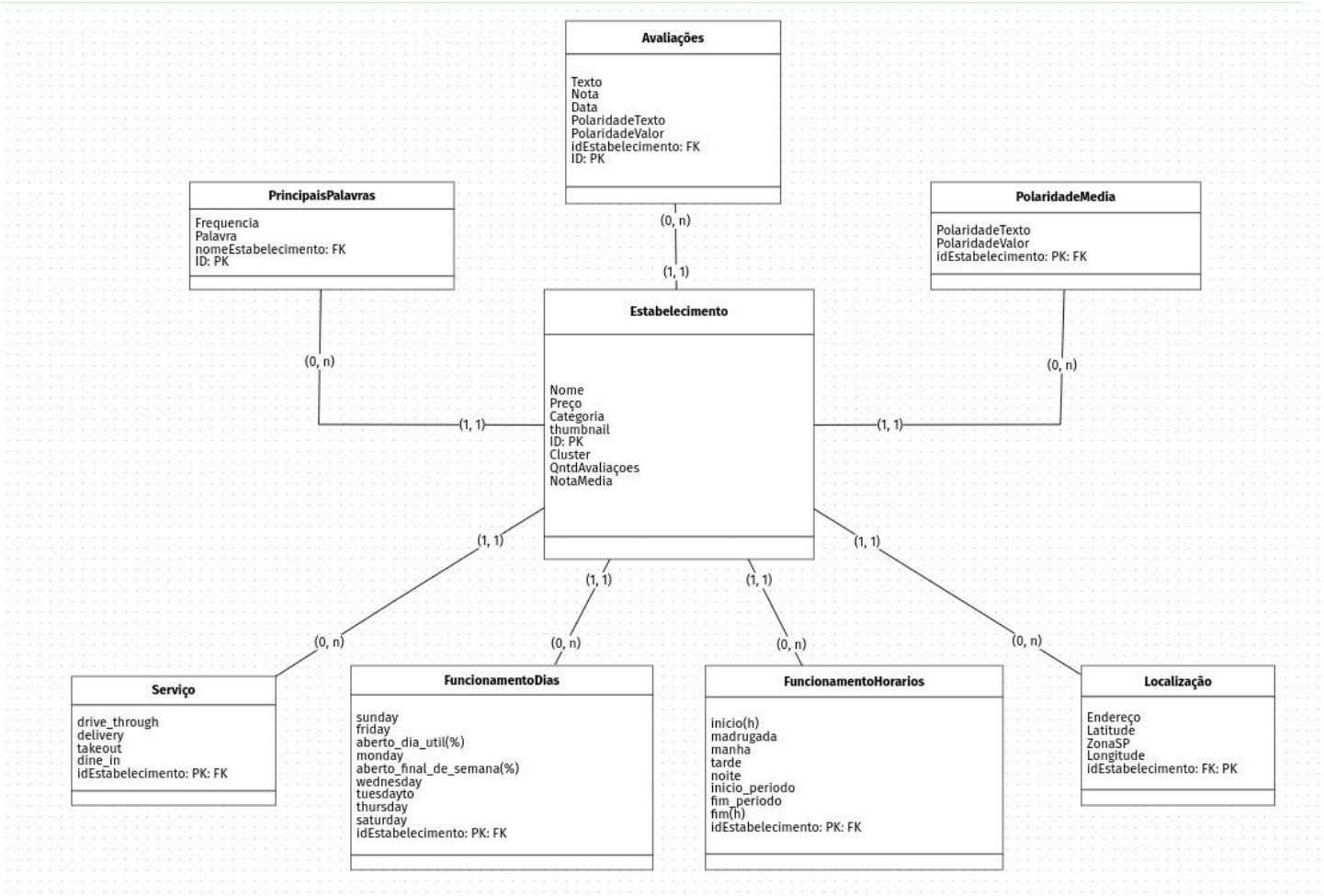
O trabalho não poderia ter saído completo se não tivéssemos uma documentação e controle de progresso, essencial para ter uma visualização clara do objetivo e os passos que devem ser realizados para alcançá-lo. Isso garante a diminuição de tensão em trabalhos em grupo, maior clareza das tarefas de cada um e mais tranquilidade com o prazo. Além disso, a organização final de todos os materiais neste caderno Jupyter é essencial para que todos os dados e informações possam vir a ser analisadas e interpretadas do material produzido por todos, agregando da forma mais clara possível o leitor.

ESTRUTURA DO BANCO DE DADOS SQL GERADO

A criação de um modelo relacional é de extrema importância para a estruturação de um banco de dados SQL. Isso porque um modelo relacional permite organizar os dados de maneira clara e eficiente, definindo as tabelas e os relacionamentos entre elas de forma lógica e coerente. Dessa forma, conseguimos garantir a integridade dos nossos dados, evitando redundâncias e inconsistências, e facilitar a consulta e a manipulação das informações armazenadas.

Sendo assim construímos um modelo relacional, com uma tabela chamada "estabelecimento" que tem relacionamentos com outras tabelas como "PrincipaisPalavras", "Avaliações", "PolaridadeMedia", "Servicos", "FuncionamentoDias", "FuncionamentoHorarios" e "Localizacao". Indicando que estamos criando um banco de dados para armazenar informações sobre estabelecimentos.

Toda tabela tem a mesma cardinalidade (1,1) e (0,n), o que indica que um estabelecimento sempre haverá no máximo uma informação da tabela relacionada, porém a tabela relacionada pode não ter nenhum estabelecimento ou pode ter vários.



Análise exploratória

A análise exploratória dos dados das API's do Google Maps Reviews e Google Maps Place Results é crucial para **compreender as características dos estabelecimentos presentes na plataforma**. Através da análise de diversos datasets, é possível obter insights valiosos sobre os horários de operação, informações gerais, opções de serviço ofertados e horários de movimento dos estabelecimentos, permitindo que nosso trabalho tenha um planejamento e organização voltado as informações de maior relevância.

Ao explorar esses dados, é possível *identificar tendências, padrões e correlações que podem ser usados para obter uma compreensão mais profunda das necessidades e comportamentos dos clientes em relação aos estabelecimentos e serviços que eles procuram*. Isso pode levar a **insights valiosos** sobre as preferências do público-alvo, concorrência, e até mesmo possíveis oportunidades de negócios e escolhas dos usuários.

Em resumo, a análise exploratória dos dados das API's do Google Maps Reviews e Google Maps Place Results, com bibliotecas de gráficos e mapas, é **fundamental para o entendimento geral do problema e identificação dos principais pontos a serem utilizados na modelagem, dashboard e até API**.

Bibliotecas

```python
import pandas as pd
import plotly.express as px
import folium
import plotly.graph_objs as go
import re
pd.set_option('display.max_columns', 30) # mostrar todas as colunas no terminal
```

MagicPython

Importando dataframes

```python
# dataset sobre horários de operação de diversos estabelecimentos
df_op = pd.read_csv('df_operating_hours.csv', sep=',')
```

MagicPython

```python
df_op.head()
```

MagicPython

	title	place_id	sunday	monday	tuesday	wednesday	thursday	friday	saturday
0	Sagrado Almoço	ChIJU008MZNTzpQRpjpsv33pRQs	Closed	8 am–8 pm	8 am–8 pm	8 am–8 pm	8 am–8 pm	8 am–8 pm	8 am–8 pm
1	2 L's Restaurante E Bar Almoço-Lanches	ChIJU4W1hWVezpQRb7UZOCXHQNk	Closed	11:30 am–2:30 pm	11:30 am–2:30 pm	11:30 am–2:30 pm	11:30 am–2:30 pm	11:30 am–2:30 pm	11:30 am–2:30 pm
2	Brasiliano Restaurante	ChIJhbffCo6KxZQREb1Ryt2u80Y	Closed	11 am–2:30 pm	11 am–2:30 pm	11 am–2:30 pm	11 am–2:30 pm	11 am–2:30 pm	11 am–2:30 pm
3	Sirva-se Almoço	ChIJzfUSXO74zpQRhxWMHFXJ9-8	Closed	11 am–3 pm	11 am–3 pm	11 am–3 pm	11 am–3 pm	11 am–3 pm	Closed
4	Speed Almoço	ChIJR5o7uexWzpQR4NL-w4Fybvg	Closed	11 am–10 pm	11 am–10 pm	11 am–10 pm	11 am–10 pm	11 am–10 pm	11 am–10 pm

```python
# dataset sobre informações gerais de diversos estabelecimentos
df_ov = pd.read_csv('df_overview.csv', sep=',')
```

MagicPython

```python
df_ov.head()
```

MagicPython

position	title	place_id	data_id	data_cid	rating	reviews	price	type	address

	position	title	place_id	data_id	data_cid	rating	reviews	price	type	addres
0	1	Sagrado Almoço	ChIJU008MZNTzpQRpjpsv33pRQs	0x94ce5393313c4d53:0xb45e97dbf6c3aa6	812312034101967526	4.6	14.0	NaN	Restaurant	Estr. d Campo Limpo, 99 - Vila Pre São Pau.
1	2	2 L's Restaurante E Bar Almoço-Lanches	ChIJU4W1hWVezpQRb7UZOCXHQNk	0x94ce5e6585b58553:0xd940c7253819b56f	15654731267408770415	4.6	17.0	NaN	Restaurant	Rua Corone Marque 399 - Vil Nov Mancheste.
2	3	Brasiliano Restaurante	ChIJhbffCo6KxZQREb1Ryt2u80Y	0x94c58a8e0adfb785:0x46f3aeddca51bd11	5112622269601004817	4.5	881.0	£	Buffet restaurant	Av. Pro Arthu Fonseca 841 Jardir Faculda.
3	4	Sirva-se Almoço	ChIJzfUSXO74zpQRhxWMHFXJ9-8	0x94cef8ee5c12f5cd:0xeff7c9551c8c1587	17291510661700654471	4.5	201.0	NaN	Buffet restaurant	R Conselhei Ribas, 330 Vil Anastácio Sã.
4	5	Speed Almoço	ChIJR5o7uexWzpQR4NL-w4Fybvg	0x94ce56ecb93b9a47:0xf86e7281c3fed2e0	17901371470508905184	4.5	26.0	NaN	Restaurant	Rua D Sílvio Dant Bertacch 253 - Vil Son.

```
# dataset sobre opções de serviço ofertados de diversos estabelecimentos
df_so = pd.read_csv('df_service_options.csv', sep=',')
```

MagicPython

```
df_so.head()
```

MagicPython

	title	place_id	dine_in	takeout	delivery	drive_through
0	Sagrado Almoço	ChIJU008MZNTzpQRpjpsv33pRQs	True	True	False	False
1	2 L's Restaurante E Bar Almoço-Lanches	ChIJU4W1hWVezpQRb7UZOCXHQNk	True	True	True	False
2	Brasiliano Restaurante	ChIJhbffCo6KxZQREb1Ryt2u80Y	True	True	False	False
3	Sirva-se Almoço	ChIJzfUSXO74zpQRhxWMHFXJ9-8	True	True	True	False
4	Speed Almoço	ChIJR5o7uexWzpQR4NL-w4Fybvg	True	True	False	False

```
# dataset sobre horários de movimento de diversos estabelecimentos
df_hm = pd.read_csv('horarios_movimento.csv', sep=',')
```

MagicPython

```
df_hm.head()
```

MagicPython

	sunday	monday	tuesday	wednesday	thursday	friday	saturday	title

	sunday	monday	tuesday	wednesday	thursday	friday	saturday	title
0	{'time': '6\u202fAM', 'busyness_score': 0}	{'time': '6\u202fAM', 'busyness_score': 0}	{'time': '6\u202fAM', 'busyness_score': 0}	{'time': '6\u202fAM', 'busyness_score': 0}	{'time': '6\u202fAM', 'busyness_score': 0}	{'time': '6\u202fAM', 'busyness_score': 0}	{'time': '6\u202fAM', 'busyness_score': 0}	Por um Punhado de Dólares
1	{'time': '7\u202fAM', 'busyness_score': 0}	{'time': '7\u202fAM', 'busyness_score': 0}	{'time': '7\u202fAM', 'busyness_score': 0}	{'time': '7\u202fAM', 'busyness_score': 0}	{'time': '7\u202fAM', 'busyness_score': 0}	{'time': '7\u202fAM', 'busyness_score': 0}	{'time': '7\u202fAM', 'busyness_score': 0}	Por um Punhado de Dólares
2	{'time': '8\u202fAM', 'busyness_score': 0}	{'time': '8\u202fAM', 'busyness_score': 0}	{'time': '8\u202fAM', 'busyness_score': 0}	{'time': '8\u202fAM', 'busyness_score': 0}	{'time': '8\u202fAM', 'busyness_score': 0}	{'time': '8\u202fAM', 'busyness_score': 0}	{'time': '8\u202fAM', 'busyness_score': 0}	Por um Punhado de Dólares
3	{'time': '9\u202fAM', 'busyness_score': 0}	{'time': '9\u202fAM', 'busyness_score': 0}	{'time': '9\u202fAM', 'busyness_score': 0}	{'time': '9\u202fAM', 'busyness_score': 0}	{'time': '9\u202fAM', 'busyness_score': 0}	{'time': '9\u202fAM', 'busyness_score': 0}	{'time': '9\u202fAM', 'busyness_score': 0}	Por um Punhado de Dólares
4	{'time': '10\u202fAM', 'info': 'Usually not to...	{'time': '10\u202fAM', 'info': 'Usually not to...	{'time': '10\u202fAM', 'info': 'Usually not to...	{'time': '10\u202fAM', 'info': 'Usually not to...	{'time': '10\u202fAM', 'info': 'Usually not to...	{'time': '10\u202fAM', 'info': 'Usually not to...	{'time': '10\u202fAM', 'info': 'Usually not to...	Por um Punhado de Dólares

```
# dataset sobre estabelecimentos relacionados com informações gerais
df_tp = pd.read_csv('tambem_procuram.csv', sep=',')
```

MagicPython

```
df_tp.head()
```

MagicPython

	estabelecimento_referencia	position	title	rating	reviews	type	thumbnail	latitude	longitude
0	Por um Punhado de Dólares	1	YERBA + Por um Punhado de Dólares	4.8	931	['Coffee shop', 'Bar', 'Book store', 'Pub', 'R...	https://lh5.googleusercontent.com/p/AF1QipO7UG...	-23.534373	-46.652746
1	Por um Punhado de Dólares	2	Urbe Cafe Bar	4.5	5879	['Coffee store', 'Bar', 'Brunch', 'Coffee shop']	https://lh5.googleusercontent.com/p/AF1QipOGuR...	-23.555976	-46.658378
2	Por um Punhado de Dólares	3	KOF - King of The Fork	4.5	1945	['Coffee store', 'Breakfast', 'Coffee shop']	https://lh5.googleusercontent.com/p/AF1QipMeeB...	-23.564052	-46.683767
3	Por um Punhado de Dólares	4	Café Girondino	4.5	6548	['Cafe', 'Breakfast', 'Cafeteria', 'Cocktail b...	https://lh5.googleusercontent.com/p/AF1QipO9py...	-23.544683	-46.634228
4	Por um Punhado de Dólares	5	Barões do Cafe	4.1	122	['Coffee shop']	https://lh5.googleusercontent.com/p/AF1QipMT6j...	-23.557180	-46.663046

Organizando datasets

```
# juntando os dataset que contem a coluna "place_id" para termos as informações de estabelecimentos em um único lugar
df_junto = pd.merge(df_op, df_ov, on='place_id', how='outer')
df_estabelecimentos = pd.merge(df_junto, df_so, on='place_id', how='outer')
```

MagicPython

```
df_estabelecimentos = df_estabelecimentos.drop(['title_x', 'title_y'], axis=1)
df_estabelecimentos.head()
```

MagicPython

	place_id	sunday	monday	tuesday	wednesday	thursday	friday	saturday	position	data_id

	place_id	sunday	monday	tuesday	wednesday	thursday	friday	saturday	position	data_id	
0	ChIJU008MZNTzpQRpjpsv33pRQs	Closed	8 am–8 pm	8 am–8 pm	8 am–8 pm	8 am–8 pm	8 am–8 pm	8 am–8 pm	1	0x94ce5393313c4d53:0xb45e97dbf6c3aa6	812
1	ChIJU4W1hWVezpQRb7UZOCXHQNk	Closed	11:30 am–2:30 pm	11:30 am–2:30 pm	11:30 am–2:30 pm	11:30 am–2:30 pm	11:30 am–2:30 pm	11:30 am–2:30 pm	2	0x94ce5e6585b58553:0xd940c7253819b56f	15654
2	ChIJhbffCo6KxZQREb1Ryt2u80Y	Closed	11 am–2:30 pm	11 am–2:30 pm	11 am–2:30 pm	11 am–2:30 pm	11 am–2:30 pm	11 am–2:30 pm	3	0x94c58a8e0adfb785:0x46f3aeddca51bd11	5112
3	ChIJzfUSXO74zpQRhxWMHFXJ9-8	Closed	11 am–3 pm	11 am–3 pm	11 am–3 pm	11 am–3 pm	11 am–3 pm	Closed	4	0x94cef8ee5c12f5cd:0xeff7c9551c8c1587	17291
4	ChIJR5o7uexWzpQR4NL-w4Fybvg	Closed	11 am–10 pm	11 am–10 pm	11 am–10 pm	11 am–10 pm	11 am–10 pm	11 am–10 pm	5	0x94ce56ecb93b9a47:0xf86e7281c3fed2e0	17901

```
# df_estabelecimentos.to_csv('info_estabelecimentos.csv', index=False)
```

MagicPython

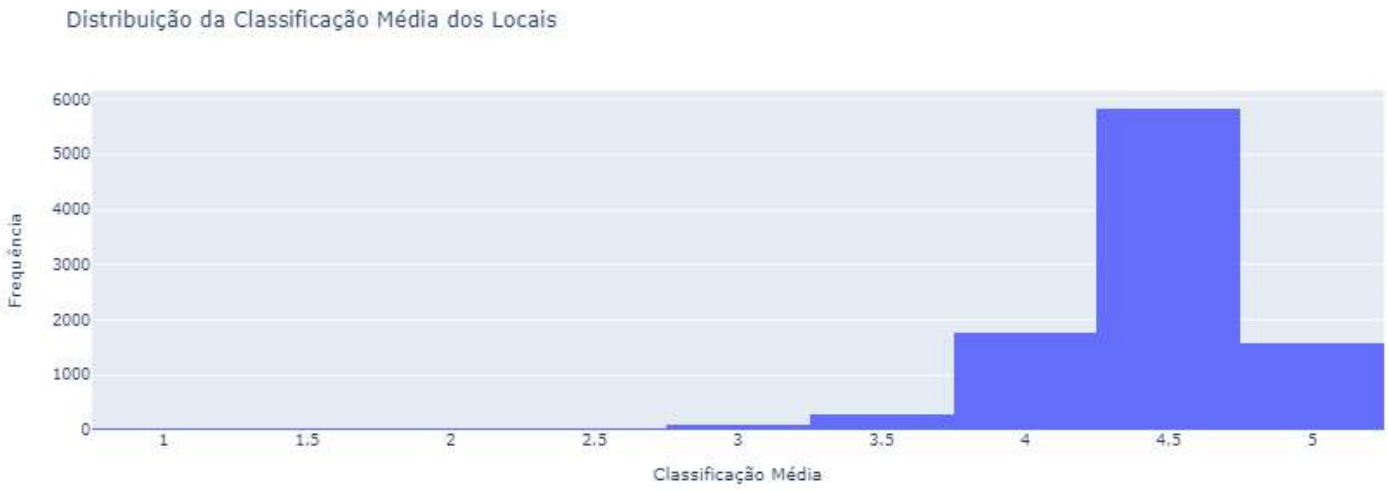
Analisando dados e informações

Dataset informações gerais do estabelecimento

```
# Plotar um histograma da coluna 'rating'
fig = go.Figure(
    [go.Histogram(
        x=df_estabelecimentos['rating'],
        nbinsx=10
    )])

fig.update_layout(
    title='Distribuição da Classificação Média dos Locais',
    xaxis_title='Classificação Média',
    yaxis_title='Frequência')

fig.show()
```

MagicPython

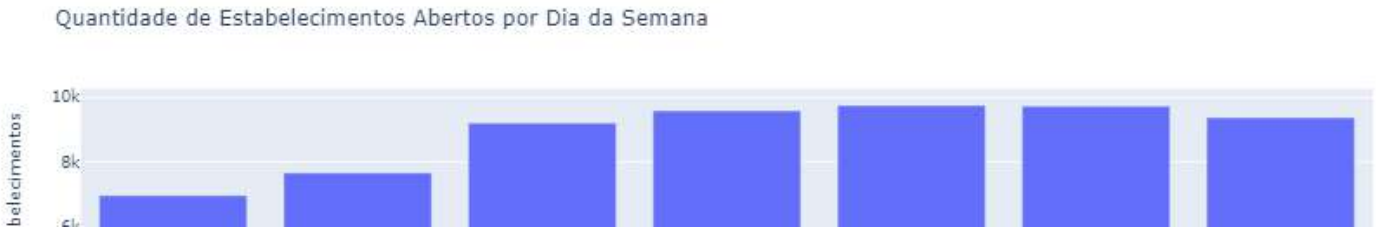
Distribuição da Classificação Média dos Locais



O histograma mostra que a maioria dos locais tem uma classificação média entre 4.3 e 4.7, o que indica que a qualidade dos locais é geralmente bem avaliada pelos clientes. Há também uma pequena quantidade de locais com classificação média abaixo de 3.7, o que indica que temos em nossa base poucos estabelecimentos com problemas de qualidade.

```python
# converte os dias da semana para uma lista
dias_da_semana = ['sunday', 'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday']

# cria um dicionário para armazenar a soma da quantidade de restaurantes abertos por dia da semana
quantidade_por_dia = {}
for dia in dias_da_semana:
    quantidade_por_dia[dia] = len(df_estabelecimentos[df_estabelecimentos[dia] != 'Closed'])

# cria o gráfico de barras
fig = go.Figure(
    [go.Bar(
        x=dias_da_semana,
        y=list(quantidade_por_dia.values())
    )])

fig.update_layout(
    title='Quantidade de Estabelecimentos Abertos por Dia da Semana',
    xaxis_title='Dia da Semana',
    yaxis_title='Quantidade de Estabelecimentos',
    xaxis_tickangle=-30)

fig.show()
```

MagicPython

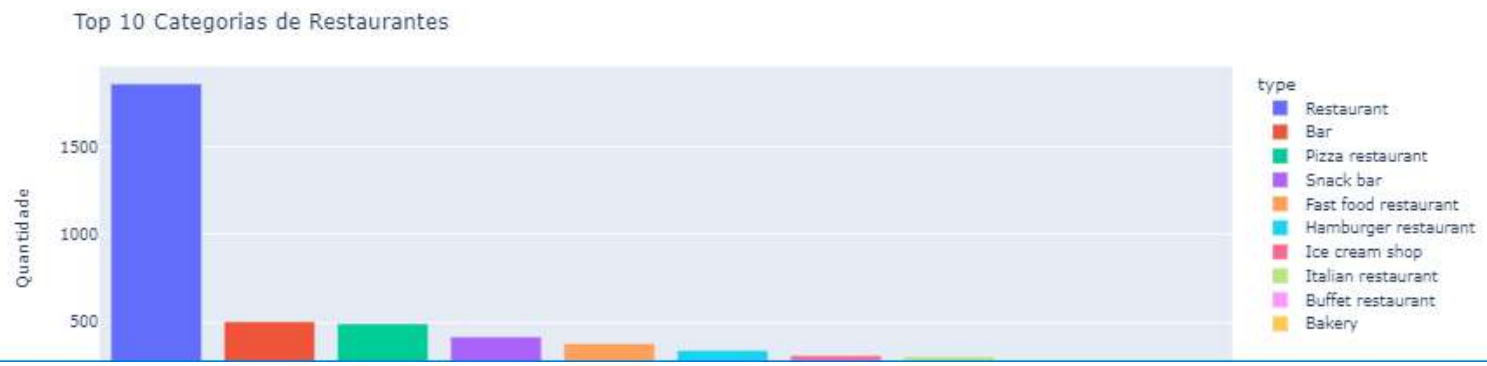


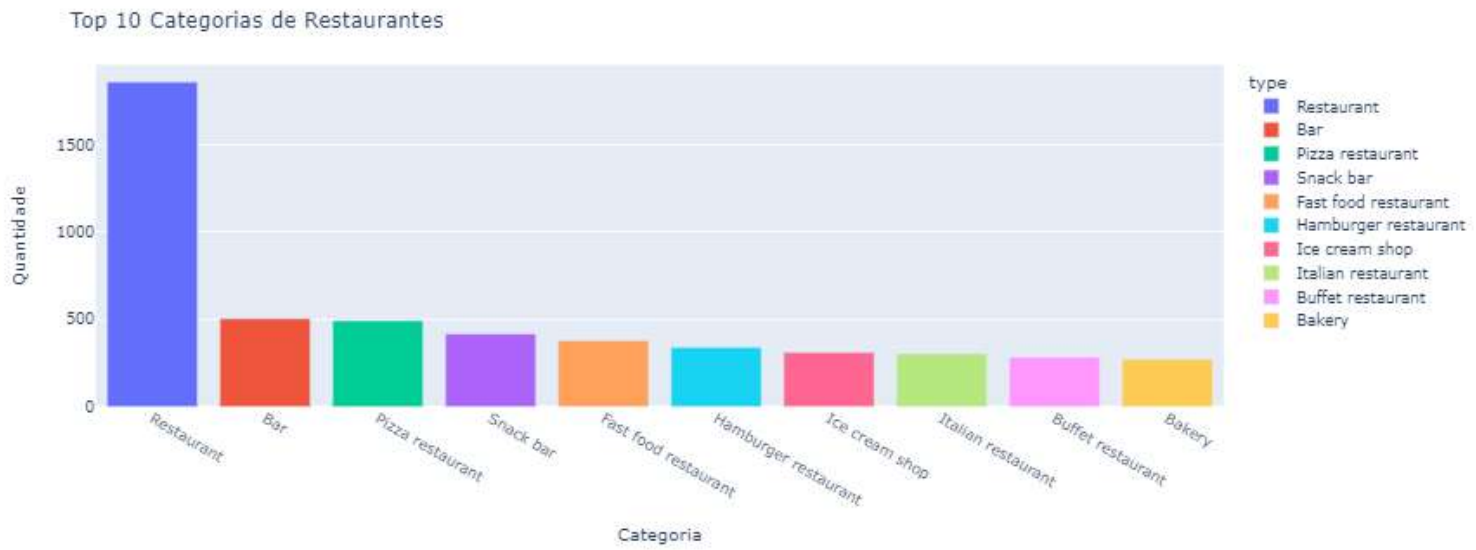
O gráfico de barras mostra a quantidade de estabelecimentos abertos por dia da semana, revelando que quarta-feira, quinta-feira e sexta-feira são os dias em que a maioria dos estabelecimentos estão abertos, enquanto domingo é o dia em que menos estabelecimentos estão abertos, provavelmente para descanso dos funcionários, ou ainda, para reduzir custos com energia elétrica e outros gastos fixos pelo menor movimento em comparação aos outros dias. Isso pode ser útil para ajudar a planejar atividades de alimentação fora de casa e para entender melhor os hábitos dos estabelecimentos.

+ Code + Markdown

```
# Agrupar por categoria e contar a quantidade de ocorrências
frequencia_tipo = df_estabelecimentos.groupby('type').size().reset_index(name='quantidade')

# Ordenar em ordem decrescente pela quantidade
frequencia_tipo = frequencia_tipo.sort_values(by='quantidade', ascending=False)

# Selecionar apenas as 5 categorias com maior frequência
top_10 = frequencia_tipo.iloc[:10]

# Criar o gráfico de barras
fig = px.bar(top_10, x='type', y='quantidade', color='type')

# Configurar o layout do gráfico
fig.update_layout(title='Top 10 Categorias de Restaurantes', xaxis_title='Categoria', yaxis_title='Quantidade')

# Exibir o gráfico
fig.show()
```

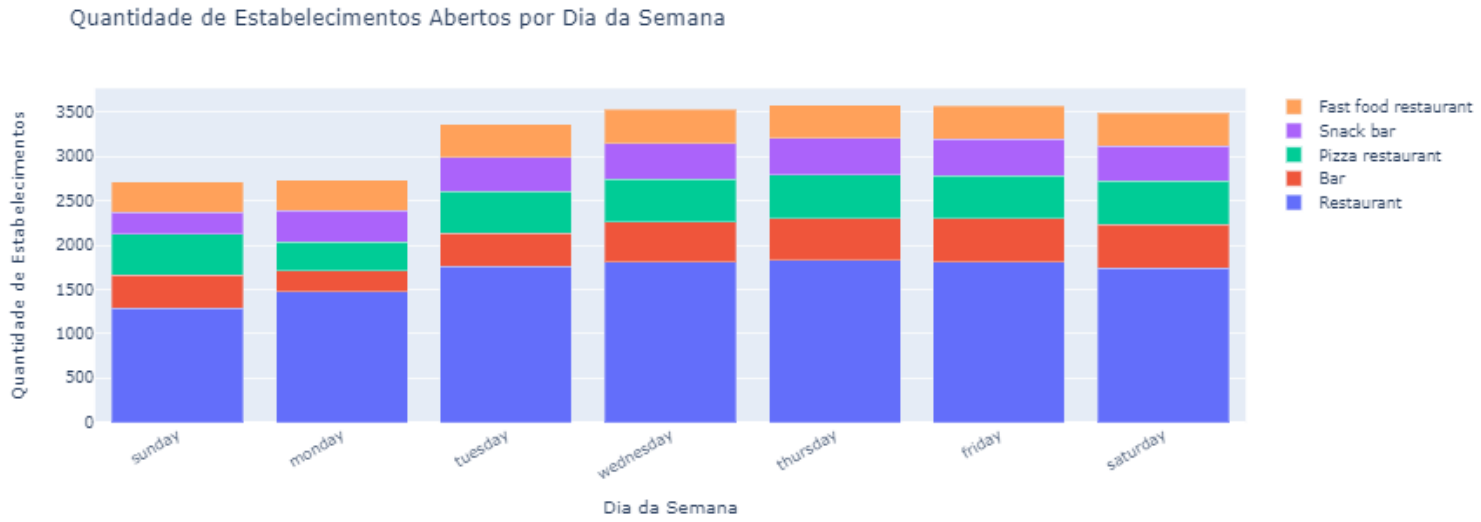
MagicPython






```python
# Selecionar os tipos de estabelecimentos a serem analisados
tipos_selecionados = ['Restaurant', 'Bar', 'Pizza restaurant', 'Snack bar', 'Fast food restaurant']

# Criar um dicionário para armazenar a quantidade de estabelecimentos abertos por dia da semana para cada tipo
dados_por_tipo = {}
for tipo in tipos_selecionados:
    df_tipo = df_estabelecimentos[df_estabelecimentos['type'] == tipo]
    quantidade_por_dia = {}
    for dia in dias_da_semana:
        quantidade_por_dia[dia] = len(df_tipo[df_tipo[dia] != 'Closed'])
    dados_por_tipo[tipo] = quantidade_por_dia

# Criar o gráfico de barras empilhadas
fig = go.Figure()
for tipo in tipos_selecionados:
    fig.add_trace(go.Bar(
        x=dias_da_semana,
        y=list(dados_por_tipo[tipo].values()),
        name=tipo
    ))

fig.update_layout(
    title='Quantidade de Estabelecimentos Abertos por Dia da Semana',
    xaxis_title='Dia da Semana',
    yaxis_title='Quantidade de Estabelecimentos',
    barmode='stack',
    xaxis_tickangle=-30
)

fig.show()
```

MagicPython



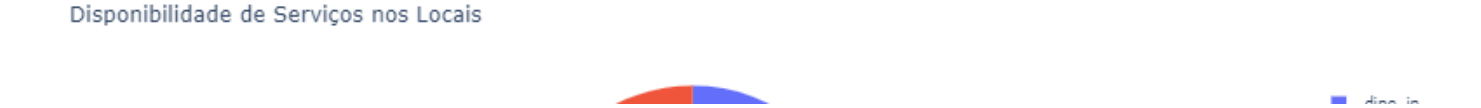
Para analisar melhor a abertura dos estabelecimentos por categoria, vemos nesse primeiro gráfico de barras as top 10 categorias que mais possuem estabelecimentos cadastrados no nosso dataset, entre els restaurantes, bar, pizzaria, snack bar, fast food, hamburgueria, sorveteria, restaurante italiano, buffet e padaria. Levando em conta as 5 top categorias podemos observar que a maioria dos restaurantes abre às quintas-feiras e que os domingos são os dias com menos estabelecimentos abertos. Já para bares, a maior quantidade de estabelecimentos abertos é aos sábados, enquanto as segundas-feiras possuem a menor quantidade. Nas pizzarias, sábados também são o dia com mais estabelecimentos abertos e as segundas-feiras possuem a menor quantidade. Para os snack bars, quintas-feiras têm a maior quantidade de estabelecimentos abertos e as segundas-feiras possuem a menor quantidade. Por fim, fast foods têm a maior quantidade de estabelecimentos abertos às quintas-feiras e a menor quantidade aos domingos.

Esses resultados podem ser explicados por ser comum que as pessoas saiam para jantar ou beber em bares e restaurantes às quintas-feiras, após um dia de trabalho, enquanto nos domingos muitas pessoas preferem ficar em casa com a família. Já aos sábados, pode haver maior procura por estabelecimentos de entretenimento noturno, o que explicaria a maior quantidade de bares e pizzarias abertas neste dia. Em geral, essas informações podem ser úteis para os proprietários desses estabelecimentos, permitindo que eles possam otimizar seus horários de funcionamento para atender melhor à demanda de seus clientes.

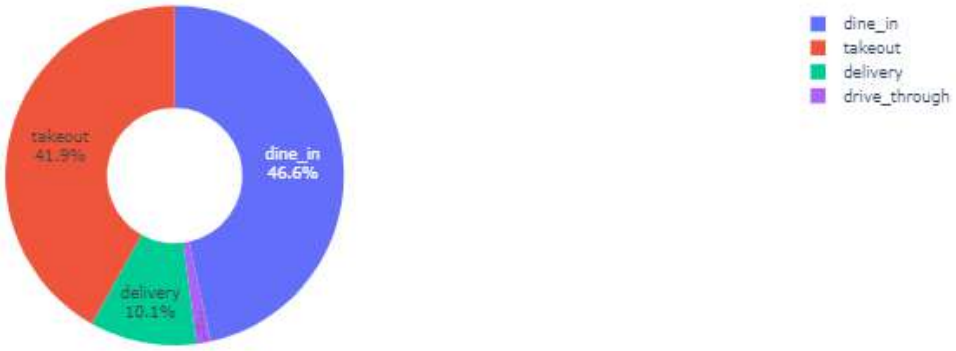
```python
# calcula a frequência de cada serviço
freq_servicos = df_estabelecimentos[['dine_in', 'takeout', 'delivery', 'drive_through']].sum()

# cria o gráfico de pizza com a frequência de cada serviço
fig = go.Figure(
    go.Pie(
        labels=freq_servicos.index.tolist(),
        values=freq_servicos.values.tolist(),
        hole=0.4,
        textposition='inside',
        textinfo='percent+label'
    ))

fig.update_layout(
    title='Disponibilidade de Serviços nos Locais',)

fig.show()
```

MagicPython

...



Disponibilidade de Serviços nos Locais



Podemos observar que a maioria dos estabelecimentos disponibilizam o serviço de "dine in" (46,6%), seguido do serviço "takeout" (41,9%). O serviço de "delivery" é oferecido em apenas 10,1% dos locais e o serviço de "drive through" é o menos comum, disponível em somente 1,36% dos estabelecimentos. Essas informações podem ser úteis para entender as preferências dos consumidores e ajudar os empreendedores a decidir quais serviços oferecer em seus estabelecimentos

```python
# selecionar as colunas de interesse
df_servicos = df_estabelecimentos[['type', 'dine_in', 'takeout', 'delivery', 'drive_through']]

# agrupar por tipo de estabelecimento e calcular a frequência de cada serviço
freq_servicos_por_tipo = df_servicos.groupby('type').sum()

# calcular a porcentagem de cada serviço para cada tipo de estabelecimento
porcentagem_servicos_por_tipo = (freq_servicos_por_tipo.div(freq_servicos_por_tipo.sum(axis=1), axis=0) * 100).round(2)

# mostrar a tabela de porcentagem de serviços por tipo de estabelecimento
print(porcentagem_servicos_por_tipo)
```

MagicPython

```
                          dine_in  takeout  delivery  drive_through
type
Adult entertainment club      NaN      NaN       NaN            NaN
Advertising agency            NaN      NaN       NaN            NaN
African restaurant          50.00    50.00      0.00            0.0
American restaurant         43.48    34.78     13.04            8.7
Andalusian restaurant      100.00     0.00      0.00            0.0
...                           ...      ...       ...            ...
Wine store                  33.33    33.33     33.33            0.0
Wine wholesaler and importer 50.00    0.00      50.00            0.0
Yakisoba Restaurant         50.00    50.00      0.00            0.0
Yakitori restaurant         50.00    50.00      0.00            0.0
Zoo                           NaN      NaN       NaN            NaN

[327 rows x 4 columns]
```

Agora exploramos mais a porcentagem presença de cada tipo de restaurante em cada segmento, e é notável que dine in e takeout são muito presentes na maioria das categorias, principalmente aquelas que envolvem restaurantes e lojas ao contrário de delivery e drive through que se concentram em segmentos mais de fast food, comida americana e lojas de bebidas

```python
# Contar valores nulos e não nulos na coluna de descrição
print(df_estabelecimentos['description'].isnull().value_counts())
```

MagicPython

```
True     7681
False    2319
Name: description, dtype: int64
```

A grande maioria dos estabelecimentos (aproximadamente 76,81%) não possui descrição disponível nessa coluna. Já os outros 23,19% possuem algum tipo de descrição cadastrada. É importante considerar que a falta de descrição pode dificultar a busca e seleção de estabelecimentos pelos clientes, além de limitar a informação disponível para análise.

```python
# Contar valores nulos e não nulos na coluna de endereço
print(df_estabelecimentos['address'].isnull().value_counts())
```

MagicPython

```
False    9936
True       64
Name: address, dtype: int64
```

A grande maioria dos estabelecimentos (99,36%) possui informações de endereço preenchidas, enquanto apenas 0,64% dos estabelecimentos (0,64%) não possuem essa informação. Podemos pensar que essa minoria que não possui endereço sejam estabelecimentos online que não possuem nenhum tipo de endereço físico.

```python
df_estabelecimentos = df_estabelecimentos.dropna(subset=['latitude', 'longitude'])

# Criar um mapa com a localização dos locais
mapa = folium.Map(location=[df_estabelecimentos['latitude'].mean(), df_estabelecimentos['longitude'].mean()], zoom_start=12)

for i, row in df_estabelecimentos.iterrows():
    folium.Marker(location=[row['latitude'], row['longitude']]).add_to(mapa)

# Exibir o mapa
mapa
```

MagicPython

Mapa com todos os estabelecimentos e suas localizações para explorar regiões e estabelecimentos pelo mapa de São Paulo

Dataset sobre os horários de movimento

Ajustando dataset para ter somente o valor do busyness_score de cada restaurante com seu nome

```python
# Selecionar as colunas desejadas
colunas_dias_semana = ['sunday', 'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday']
df_dias_semana = df_hm[colunas_dias_semana]

# Definir a função para extrair o valor do busyness_score
def extrair_score(coluna):
    # Extrair o valor usando regex
    match = re.search("'busyness_score':\s*(\d+)", str(coluna))
    if match:
        return match.group(1)
    else:
        return None

# Aplicar a função a todas as colunas selecionadas
```

```python
# Aplicar a função a todas as colunas selecionadas
df_dds = df_dias_semana.apply(lambda x: x.apply(extrair_score))

# Concatenar as colunas em um novo dataframe
df_final = pd.concat([df_hm['title'], df_dds], axis=1)

# Exibir o resultado
df_final
```

MagicPython

	title	sunday	monday	tuesday	wednesday	thursday	friday	saturday
0	Por um Punhado de Dólares	0	0	0	0	0	0	0
1	Por um Punhado de Dólares	0	0	0	0	0	0	0
2	Por um Punhado de Dólares	0	0	0	0	0	0	0
3	Por um Punhado de Dólares	0	0	0	0	0	0	0
4	Por um Punhado de Dólares	27	39	29	34	34	31	29
...	...	...	...	...	...	...	...	...
5399	Emporio Akkar	None	0	0	0	0	0	0
5400	Emporio Akkar	None	0	0	0	0	0	0
5401	Emporio Akkar	None	0	0	0	0	0	0
5402	Emporio Akkar	None	0	0	0	0	0	0
5403	Emporio Akkar	None	0	0	0	0	0	0

5404 rows × 8 columns

```python
# Selecionar apenas as linhas com valores acima de zero e não nulos
df_filtrado = df_final.loc[(df_final[colunas_dias_semana].astype(float) > 0).any(axis=1) & df_final[colunas_dias_semana].notnull().all(axis=1)]

# Exibir o resultado
df_filtrado.head()
```

MagicPython

	title	sunday	monday	tuesday	wednesday	thursday	friday	saturday
4	Por um Punhado de Dólares	27	39	29	34	34	31	29
5	Por um Punhado de Dólares	45	51	44	48	51	44	47
6	Por um Punhado de Dólares	64	63	58	60	69	55	65
7	Por um Punhado de Dólares	80	72	70	69	81	63	77
8	Por um Punhado de Dólares	90	78	76	72	81	65	86

Dataset sobre relações entre estabelecimentos

```python
df_tp
```

MagicPython

	estabelecimento_referencia	position	title	rating	reviews	type	thumbnail	latitude	longitude
0	Por um Punhado de Dólares	1	YERBA + Por um Punhado de Dólares	4.8	931	['Coffee shop', 'Bar', 'Book store', 'Pub', 'R...	https://lh5.googleusercontent.com/p/AF1QipO7UG...	-23.534373	-46.652746
1	Por um Punhado de Dólares	2	Urbe Cafe Bar	4.5	5879	['Coffee store', 'Bar', 'Brunch', 'Coffee shop']	https://lh5.googleusercontent.com/p/AF1QipOGuR...	-23.555976	-46.658378

1834	Emporio Akkar	1	Empório Syrio	4.6	428	['Gourmet grocery store', 'Market']	https://lh5.googleusercontent.com/p/AF1QipNV1X...	-23.543061	-46.631866
1835	Emporio Akkar	2	Tio Ali Empório Árabe	4.8	33	['Gourmet grocery store']	https://lh5.googleusercontent.com/p/AF1QipMPFT...	-23.541785	-46.629380
1836	Emporio Akkar	3	Maxifour Lebanon Market Center - Empório Árabe	4.4	1155	['Gourmet grocery store', 'Condiments supplier...	https://lh5.googleusercontent.com/p/AF1QipNz6S...	-23.613913	-46.658768
1837	Emporio Akkar	4	Raful Cozinha Árabe - 25 de Março	4.6	7929	['Middle Eastern', 'Sfiha restaurant']	https://lh5.googleusercontent.com/p/AF1QipOEhq...	-23.543192	-46.631870
1838	Emporio Akkar	5	Faruk Arab sweets	4.6	228	['Candy store']	https://lh5.googleusercontent.com/p/AF1QipMDx0...	-23.540655	-46.604723

1839 rows × 9 columns

Você pode usar esse dataset, para filtrar restaurante e achar suas similaridades com os outros estabelecimentos.

Modelo de NPL com Análise de sentimento

O modelo de NLP (Processamento de Linguagem Natural) que analisa os sentimentos das avaliações de diversos estabelecimentos é uma técnica de análise de dados que utiliza algoritmos de aprendizado de máquina para compreender o tom e o contexto das avaliações deixadas pelos clientes na plataforma do Google.

Ao analisar as avaliações dos clientes, o modelo de NLP pode identificar tendências e padrões nos sentimentos expressos pelos clientes, permitindo que as empresas e usuários obtenham insights valiosos sobre o desempenho de seus estabelecimentos e serviços.

Em resumo, o modelo de NLP que analisa os sentimentos das avaliações de diversos estabelecimentos é uma técnica poderosa de análise de dados que pode fornecer informações valiosas, em nosso caso mais para clientes que podem utilizar das análises para observar a verdadeira opinião além das notas/estrelas, comparadas também a categorias e avaliações.

Bibliotecas

```
[185]
import pandas as pd
import nltk
from nltk.corpus import stopwords
from textblob import TextBlob
from wordcloud import WordCloud
from matplotlib import pyplot as plt
from numba import jit, prange, njit
import time
# from deep_translator import GoogleTranslator, single_detection

from sumy.nlp.tokenizers import Tokenizer
from sumy.parsers.plaintext import PlaintextParser
from sumy.summarizers.lsa import LsaSummarizer

# download de stopwords de nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('vader_lexicon')
```

MagicPython

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
```

Pré Processamento

Antes de analisar os sentimentos de cada restaurante, devemos filtrar e limpar os datasets que iremos usar

```
#Limpeza de dados
df = pd.read_csv('most_relevant_reviews.csv',sep = ',', usecols=['title', 'place_id', 'username', 'rating', 'description', 'date'])
df
```

[98] MagicPython

	title	place_id	username	rating	description	date
0	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9I10wA	Vanessa Scott	5	Very chill and hipster, cool place! Came here ...	a month ago
1	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9I10wA	Pedro Henrique Fernandes	5	Superb coffee shop in São Paulo, easily one of...	3 months ago
2	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9I10wA	Caio Cruz	5	Really cool atmosphere plus good cakes and cof...	9 months ago
3	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9I10wA	Sebastian HH	5	Super nice cafe !\nDelicious coffee and sandwi...	a month ago
4	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9I10wA	Rachel B	5	Super enjoyed this place! We came in passing w...	4 years ago
...	...	...	...	...	...	...
4331	Villares Lounge Club	ChIJV1wC_1j3zpQRCrOgpy20cOI	André Marques	5	NaN	a month ago
4332	Villares Lounge Club	ChIJV1wC_1j3zpQRCrOgpy20cOI	Richard Lucas	4	NaN	7 months ago
4333	Villares Lounge Club	ChIJV1wC_1j3zpQRCrOgpy20cOI	Iago Vilas Boas	4	NaN	2 months ago
4334	Villares Lounge Club	ChIJV1wC_1j3zpQRCrOgpy20cOI	Denis Balduino	5	NaN	3 weeks ago
4335	Villares Lounge Club	ChIJV1wC_1j3zpQRCrOgpy20cOI	João Henrique	5	NaN	3 months ago

4336 rows × 6 columns

```
#Organizar o dataset
df = df.rename(columns={"title":"restaurante"})
df = df.rename(columns={"description":"comentario"})
df = df.dropna().reset_index(drop=True)
df
```

[99] MagicPython

	restaurante	place_id	username	rating	comentario	date
0	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9I10wA	Vanessa Scott	5	Very chill and hipster, cool place! Came here ...	a month ago
1	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9I10wA	Pedro Henrique Fernandes	5	Superb coffee shop in São Paulo, easily one of...	3 months ago
2	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9I10wA	Caio Cruz	5	Really cool atmosphere plus good cakes and cof...	9 months ago
3	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9I10wA	Sebastian HH	5	Super nice cafe !\nDelicious coffee and sandwi...	a month ago
4	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9I10wA	Rachel B	5	Super enjoyed this place! We came in passing w...	4 years ago
...	...	...	...	...	...	...
4224	Pastel da Rose	ChIJsUvtjMRZzpQRIV9yNjTkbUE	Balatonfüredi Dávid	5	Best pastel in town! O maior é mais recheado!...	3 months ago
4225	Pastel da Rose	ChIJsUvtjMRZzpQRIV9yNjTkbUE	João Carlos Sousa	5	Simpática e prestativa, Rose faz pastéis na ho...	7 months ago
4226	Pastel da Rose	ChIJsUvtjMRZzpQRIV9yNjTkbUE	Isabela Pontes	5	Meu deus, o pastel é sensaaacional se bom. Mas...	9 months ago
4227	Pastel da Rose	ChIJsUvtjMRZzpQRIV9yNjTkbUE	Henrique Costa (Silva)	5	Sem duvidas o melhor pastel da cidade!! Além d...	a year ago
4228	Villares Lounge Club	ChIJV1wC_1j3zpQRCrOgpy20cOI	SAAD GHANEM	5	Lugar muito bom	3 months ago

4229 rows × 6 columns

```
# traduzindo dados para inglês, usando computação paralela
@jit(parallel=True)
def traduzir_comentarios(comentarios):
```

```python
# traduzindo dados para inglês, usando computação paralela
@jit(parallel=True)
def traduzir_comentarios(comentarios):
  translated_list = []
  my_translator = GoogleTranslator(source='auto', target='english')

  for c in prange(len(comentarios)):
    result = my_translator.translate(text=str(comentarios[c]))
    translated_list.append(result)
  return translated_list
```

[100] MagicPython

```python
tempo_inicial = time.time()
df['comentario'] = traduzir_comentarios(df.comentario.values)
tempo_final = time.time()
print(f"A duração da tradução foi de {tempo_final - tempo_inicial} segundos")
```

[106] MagicPython

```
A duração da tradução foi de 13 segundos
```

```python
#salvando no csv
df.to_csv('most_relevant_reviews_translated.csv', index=False)
```

MagicPython

Remover os stopwords

```python
df = pd.read_csv('most_relevant_reviews_translated.csv',sep=',')
df
```

[73] MagicPython

	restaurante	place_id	username	rating	comentario	date
0	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9l10wA	Vanessa Scott	5	Very chill and hipster, cool place! Came here ...	a month ago
1	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9l10wA	Pedro Henrique Fernandes	5	Superb coffee shop in São Paulo, easily one of...	3 months ago
2	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9l10wA	Caio Cruz	5	Really cool atmosphere plus good cakes and cof...	9 months ago
3	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9l10wA	Sebastian HH	5	Super nice cafe !\nDelicious coffee and sandwi...	a month ago
4	Por um Punhado de Dólares	ChIJ08g_akxYzpQR0RO2V9l10wA	Rachel B	5	Super enjoyed this place! We came in passing w...	4 years ago
...	...	...	...	...	...	...
4224	Pastel da Rose	ChIJsUvtjMRZzpQRlV9yNjTkbUE	Balatonfüredi Dávid	5	Best pastel in town! The bigger one is more st...	3 months ago
4225	Pastel da Rose	ChIJsUvtjMRZzpQRlV9yNjTkbUE	João Carlos Sousa	5	Friendly and helpful, Rose makes pastries on t...	7 months ago
4226	Pastel da Rose	ChIJsUvtjMRZzpQRlV9yNjTkbUE	Isabela Pontes	5	My god, the pastel is sensational if good. Dry...	9 months ago
4227	Pastel da Rose	ChIJsUvtjMRZzpQRlV9yNjTkbUE	Henrique Costa (Silva)	5	Without a doubt the best pastel in town!! In a...	a year ago
4228	Villares Lounge Club	ChIJV1wC_1j3zpQRCrOgpy20cOI	SAAD GHANEM	5	very good place	3 months ago

4229 rows × 6 columns

```python
# Criando DataFrames vazios para armazenar pontuações de polaridade de sentimento e frequencias de palavras
sentimento_df = pd.DataFrame(columns=['restaurante', 'comentario', 'polaridade'])
freq_df = pd.DataFrame(columns=['restaurante', 'palavra', 'frequencia'])
```

[74] MagicPython

```python
# Para cada avaliação de restaurante, análisamos a polaridade de sentimentos e a estatísticas de frequência de palavras
for restaurante in df['restaurante'].unique():
    reviews = df.loc[df['restaurante'] == restaurante, 'comentario']

    words = []
    # analise de sentimento
    for comentario in reviews:
        # Tokenização de texto e exclusão de stopword
        review_str = str(comentario)
        tokens = nltk.word_tokenize(review_str.lower())

        # calculando polaridade
        blob = TextBlob(' '.join(tokens))
        polaridade = blob.sentiment.polarity

        # Adicionando pontuações de polaridade de sentimento no DataFrame "sentimento_df"
        sentimento_df = sentimento_df.append({
            'restaurante': restaurante,
            'comentario': comentario,
            'polaridade': polaridade
        }, ignore_index=True)

        #estrutura de repetição para exclusão de stopwords em portugues e ingles
        linguagens = ['english', 'portuguese']
        tokens_freq = tokens.copy()
        for l in linguagens:
            tokens_freq = [t for t in tokens_freq if t.isalpha() and t not in stopwords.words(l)]

        words += tokens_freq

    freq_dist = nltk.FreqDist(words)

    # Adicionando pontuações de polaridade de sentimento no DataFrame "freq_df"
    for palavra, frequencia in freq_dist.items():
        freq_df = freq_df.append({
            'restaurante': restaurante,
            'palavra': palavra,
            'frequencia': frequencia
        }, ignore_index=True)
```

```
Output exceeds the size limit. Open the full output data in a text editor
A saída de streaming foi truncada nas últimas 5000 linhas.
<ipython-input-75-549617062d0b>:35: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.conc
  freq_df = freq_df.append({
<ipython-input-75-549617062d0b>:35: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.conc
  freq_df = freq_df.append({
<ipython-input-75-549617062d0b>:35: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.conc
  freq_df = freq_df.append({
<ipython-input-75-549617062d0b>:35: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.conc
  freq_df = freq_df.append({
<ipython-input-75-549617062d0b>:35: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.conc
  freq_df = freq_df.append({
<ipython-input-75-549617062d0b>:35: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.conc
  freq_df = freq_df.append({
<ipython-input-75-549617062d0b>:35: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.conc
  freq_df = freq_df.append({
...
<ipython-input-75-549617062d0b>:35: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.conc
  freq_df = freq_df.append({
<ipython-input-75-549617062d0b>:35: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.conc
  freq_df = freq_df.append({
```

```
sentimento_df['rating']=df['rating']
sentimento_df['place_id']=df['place_id']
sentimento_df
```

[76] MagicPython

	restaurante	comentario	polaridade	rating	place_id
0	Por um Punhado de Dólares	Very chill and hipster, cool place! Came here ...	0.510648	5	ChIJ08g_akxYzpQR0RO2V9l10wA
1	Por um Punhado de Dólares	Superb coffee shop in São Paulo, easily one of...	0.257292	5	ChIJ08g_akxYzpQR0RO2V9l10wA
2	Por um Punhado de Dólares	Really cool atmosphere plus good cakes and cof...	0.258333	5	ChIJ08g_akxYzpQR0RO2V9l10wA
3	Por um Punhado de Dólares	Super nice cafe !\nDelicious coffee and sandwi...	0.694444	5	ChIJ08g_akxYzpQR0RO2V9l10wA
4	Por um Punhado de Dólares	Super enjoyed this place! We came in passing w...	0.530889	5	ChIJ08g_akxYzpQR0RO2V9l10wA
...	...	...	...	...	...
4224	Pastel da Rose	Best pastel in town! The bigger one is more st...	0.562500	5	ChIJsUvtjMRZzpQRlV9yNjTkbUE
4225	Pastel da Rose	Friendly and helpful, Rose makes pastries on t...	0.458273	5	ChIJsUvtjMRZzpQRlV9yNjTkbUE
4226	Pastel da Rose	My god, the pastel is sensational if good. Dry...	0.469444	5	ChIJsUvtjMRZzpQRlV9yNjTkbUE
4227	Pastel da Rose	Without a doubt the best pastel in town!! In a...	0.603704	5	ChIJsUvtjMRZzpQRlV9yNjTkbUE
4228	Villares Lounge Club	very good place	0.910000	5	ChIJV1wC_1j3zpQRCrOgpy20cOI

4229 rows × 5 columns

```
#Criar a função para idetificar se a polaridade é positivo, neutro ou negativo
def polaridade(nota):
    if nota > 0:
        return (1, 'Positivo')
    elif nota < 0:
        return (-1, 'Negativo')
    else:
        return (0, 'Neutro')

# Adicionar tanto a polaridade em número quanto em texto em uma nova coluna
sentimento_df[['polaridade_num', 'polaridade_texto']] = sentimento_df['polaridade'].apply(polaridade).apply(pd.Series)
sentimento_df
```

[77] MagicPython

	restaurante	comentario	polaridade	rating	place_id	polaridade_num	polaridade_texto
0	Por um Punhado de Dólares	Very chill and hipster, cool place! Came here ...	0.510648	5	ChIJ08g_akxYzpQR0RO2V9l10wA	1	Positivo
1	Por um Punhado de Dólares	Superb coffee shop in São Paulo, easily one of...	0.257292	5	ChIJ08g_akxYzpQR0RO2V9l10wA	1	Positivo
2	Por um Punhado de Dólares	Really cool atmosphere plus good cakes and cof...	0.258333	5	ChIJ08g_akxYzpQR0RO2V9l10wA	1	Positivo
3	Por um Punhado de Dólares	Super nice cafe !\nDelicious coffee and sandwi...	0.694444	5	ChIJ08g_akxYzpQR0RO2V9l10wA	1	Positivo
4	Por um Punhado de Dólares	Super enjoyed this place! We came in passing w...	0.530889	5	ChIJ08g_akxYzpQR0RO2V9l10wA	1	Positivo
...	...	...	...	...	...	...	...
4224	Pastel da Rose	Best pastel in town! The bigger one is more st...	0.562500	5	ChIJsUvtjMRZzpQRlV9yNjTkbUE	1	Positivo
4225	Pastel da Rose	Friendly and helpful, Rose makes pastries on t...	0.458273	5	ChIJsUvtjMRZzpQRlV9yNjTkbUE	1	Positivo
4226	Pastel da Rose	My god, the pastel is sensational if good. Dry...	0.469444	5	ChIJsUvtjMRZzpQRlV9yNjTkbUE	1	Positivo
4227	Pastel da Rose	Without a doubt the best pastel in town!! In a...	0.603704	5	ChIJsUvtjMRZzpQRlV9yNjTkbUE	1	Positivo
4228	Villares Lounge Club	very good place	0.910000	5	ChIJV1wC_1j3zpQRCrOgpy20cOI	1	Positivo

4229 rows × 7 columns

```
# Salvando pontuações de polaridade de sentimento e frequências de palavras para um arquivo CSV
```

```
# Salvando pontuações de polaridade de sentimento e frequências de palavras para um arquivo CSV
sentimento_df.to_csv('restaurant_polaridade.csv', index=False)
freq_df.to_csv('restaurant_word_freq.csv', index=False)
```

MagicPython

Análise de sentimento

```
sent_restaurante = sentimento_df.copy()
sent_restaurante.head()
```

[78]

MagicPython

	restaurante	comentario	polaridade	rating	place_id	polaridade_num	polaridade_texto
0	Por um Punhado de Dólares	Very chill and hipster, cool place! Came here ...	0.510648	5	ChIJ08g_akxYzpQR0RO2V9I10wA	1	Positivo
1	Por um Punhado de Dólares	Superb coffee shop in São Paulo, easily one of...	0.257292	5	ChIJ08g_akxYzpQR0RO2V9I10wA	1	Positivo
2	Por um Punhado de Dólares	Really cool atmosphere plus good cakes and cof...	0.258333	5	ChIJ08g_akxYzpQR0RO2V9I10wA	1	Positivo
3	Por um Punhado de Dólares	Super nice cafe !\nDelicious coffee and sandwi...	0.694444	5	ChIJ08g_akxYzpQR0RO2V9I10wA	1	Positivo
4	Por um Punhado de Dólares	Super enjoyed this place! We came in passing w...	0.530889	5	ChIJ08g_akxYzpQR0RO2V9I10wA	1	Positivo

```
#melhores restaurantes por meio de analise de sentimento
top_ratings = sent_restaurante.sort_values(['polaridade','rating'], ascending=False)
top_ratings[:]
```

[79]

MagicPython

	restaurante	comentario	polaridade	rating	place_id	polaridade_num	polaridade_texto
62	Café Sol	Went there for coffee and a dessert. Everythin...	1.0	5	ChIJKY73HaIZzpQR1ZE9JE8dzS0	1	Positivo
68	Um Coffee Co.	Awesome!!	1.0	5	ChIJgY-iHWZYzpQRUQ1dHiOh1IU	1	Positivo
79	Um Coffee Co.	Great service and coffee!	1.0	5	ChIJy_Nx8yVazpQRf53c9meKjik	1	Positivo
155	Starbucks	Great service! And a delicious brigadeiro frap...	1.0	5	ChIJYdxZ6Z1XzpQRF6a4w5mWwzg	1	Positivo
177	the little coffee shop , CAFÉ ESPECIAL & CURSO...	One of the best in São Paulo, no doubt. Great ...	1.0	5	ChIJLZAxhsdZzpQRqTokkP9PAKg	1	Positivo
...	...	...	...	...	...	...	...
3612	Bolo da Madre	I had a terrible experience, my order was canc...	-1.0	5	ChIJMxUQOZtbzpQRMFfnopTnW0M	-1	Negativo
3932	Fábrica de Bolo Vó Alzira Bangu	I bought a passion fruit cake, through Ifood, ...	-1.0	5	ChIJm2OcUOtYzpQRvQ0INNaEdYQ	-1	Negativo
151	Starbucks	Terrible service.\nHad an argument with one of...	-1.0	4	ChIJgX_VHYNXzpQR8VLomhGDEjk	-1	Negativo
2577	Confeitaria e Panificadora Doce Belo	Just horrible this owner treats others as if t...	-1.0	4	ChIJecU9gcDNyJQRdLbIg_1Jeus	-1	Negativo
3771	Outback Steakhouse	Horrible customer service	-1.0	3	ChIJ6YQdCZyXpgARJxtWbmE5ysU	-1	Negativo

4229 rows × 7 columns

```
#Se estivermos preocupados com as avaliações totais de sentimento de diferentes restaurantes, podemos primeiro olhar para a média dessas avaliações do res
mean_ratings = sent_restaurante.pivot_table(values=['polaridade','rating'], index='restaurante',aggfunc='mean')
mean_ratings[:5]
```

[80]

MagicPython

restaurante	polaridade	rating
'Liderança MaxCoffee Quality | Locação E Venda De Máquinas De Café Expresso Profissional - Vending - Café TerraGrão - Eventos	0.588156	4.375
89ºC Coffee Station	0.409069	4.000
@ Zona Norte	0.918333	3.500
A Baianeira	0.493646	4.625
A Bolaria	0.769018	5.000

```
#Vendo  estatisticas de cada restaurante e o número de votantes:
ratings_by_place = sent_restaurante.groupby('restaurante').size()
ratings_by_place[:10]
```

[81] MagicPython

```
restaurante
'Liderança MaxCoffee Quality | Locação E Venda De Máquinas De Café Expresso Profissional - Vending - Café TerraGrão - Eventos    8
89ºC Coffee Station                                                                                                               8
@ Zona Norte                                                                                                                      2
A Baianeira                                                                                                                       8
A Bolaria                                                                                                                         7
A Casa do Porco Bar                                                                                                               8
A Creperia                                                                                                                        8
A Douceur Doces Finos                                                                                                             4
A Imperatriz • Casa de Chá - Presentes - Terapias                                                                                 8
A Quinta do Marquês- Xodó Paulista                                                                                                8
dtype: int64
```

```
#Se o número de votantes for muito baixo, então esses dados não são objetivos, vamos selecionar restaurantes com mais de 4 votantes:
active_place = ratings_by_place.index[ratings_by_place >= 5]
active_place
```

[82] MagicPython

```
Index(['''Liderança MaxCoffee Quality | Locação E Venda De Máquinas De Café Expresso Profissional - Vending - Café TerraGrão - Eventos',
       '89ºC Coffee Station', 'A Baianeira', 'A Bolaria',
       'A Casa do Porco Bar', 'A Creperia',
       'A Imperatriz • Casa de Chá - Presentes - Terapias',
       'A Quinta do Marquês- Xodó Paulista',
       'A Refinaria Gourmet Unidade Vila Leopoldina',
       'ALTO DA MOOCA PÃES E DOCES',
       ...
       'YERBA + Por um Punhado de Dólares', 'ZUO Café', 'Zeu's Hot Dog',
       'Ziriguidum Bar', 'boteco Tamandaré', 'café paris', 'café zinn',
       'padaria', 'quilinho',
       'the little coffee shop . CAFÉ ESPECIAL & CURSOS . Delivery/retiradas agendadas, e workshop online p/ quem quer empreender.'],
      dtype='object', name='restaurante', length=492)
```

```
#media de ratings
mean_ratings = mean_ratings.loc[active_place]
mean_ratings
```

[83] MagicPython

restaurante	polaridade	rating
'Liderança MaxCoffee Quality | Locação E Venda De Máquinas De Café Expresso Profissional - Vending - Café TerraGrão - Eventos	0.588156	4.375
89ºC Coffee Station	0.409069	4.000
A Baianeira	0.493646	4.625
A Bolaria	0.769018	5.000

café paris	0.427439	4.625
café zinn	0.362548	4.125
padaria	0.440074	4.000
quilinho	0.170510	4.500
the little coffee shop . CAFÉ ESPECIAL & CURSOS . Delivery/retiradas agendadas, e workshop online p/ quem quer empreender.	0.540140	4.500

492 rows × 2 columns

```
#ordenando os 10 primeiros restaurantes por meio de rating
top_ratings = mean_ratings.sort_values(by='rating', ascending=False)
top_ratings[:10]
```

[84] MagicPython

restaurante	polaridade	rating
Pastel da Rose	0.508400	5.0
Spoleto	-0.061226	5.0
Nó 8 Café	0.401394	5.0
Brothaus Breads & Pastries	0.373607	5.0
Oggi Sorvetes	0.662490	5.0
Oggi Sorvetes - Vila Guarani	0.541944	5.0
Padaria Bella Paulista	0.325796	5.0
Padaria Brasileira	0.338835	5.0
Bar do Nico	0.489397	5.0
Padaria Estrela Polar	0.327431	5.0

+ Code + Markdown

```
#ordenando os 10 primeiros restaurantes por meio de média de polaridade
top_ratings = mean_ratings.sort_values(by='polaridade', ascending=False)
top_ratings[:10]
```

[85] MagicPython

restaurante	polaridade	rating
A Bolaria	0.769018	5.000
Cafeteria São Paulo	0.742917	4.250
Tetê Confectionery Gelateria Cafe	0.724749	4.000
Farsoun Comida e Doces Árabe Sirio	0.718557	4.875
Actual - Máquinas De Café em São Paulo - Venda e Locação Máquinas de Café	0.703125	5.000
Abarista Cafés	0.693690	4.000
Do Digo Macarons e Chocolates	0.685147	4.250
The Francis Bolos e Doces	0.680933	4.625
Nakajyma Sushi Limão	0.679310	4.250
Duo - Café & Comidinhas	0.678598	4.500

Conclusão da Análise de Sentimento

O cálculo de polaridade e rating é diferente devido a polaridade foi calculada com base nas avaliações dos clientes e o rating foi baseado em critérios como

Conclusão da Análise de Sentimento

O cálculo de polaridade e rating é diferente devido a polaridade foi calculada com base nas avaliações dos clientes e o rating foi baseado em critérios como qualidade da comida, serviço e ambiente, mostrado na output de frequência de palavras. Assim, é possível que a polaridade e o rating sejam opostos. Então, a relação entre a polaridade e o rating pode variar de acordo com o contexto e as métricas específicas usadas para calcular cada um. Por isso, é difícil analisar por aqui, portanto, será mostrado mais explicativo no analise de frequência de palavras.

Analise de frequências das palavras

```
top_ratings = freq_df.sort_values(['frequencia'], ascending=False)
top_ratings[:10]
```

[86] MagicPython

	restaurante	palavra	frequencia
4928	The Coffee	coffee	99
4959	The Coffee	service	33
4961	The Coffee	place	24
4933	The Coffee	good	24
10829	Saint Decor Bistrô	place	20
44304	Casa de Bolos - Teodoro Sampaio 636 (PRÓXIMO A...	cake	20
15860	Le Pain Quotidien	good	20
674	Coffee Lab	coffee	19
4934	The Coffee	quality	19
5017	The Coffee	great	18

```
# Definir o nome do restaurante que você deseja analisar
restaurante = input(str('Digite o restaurante desejado: ')) #Nesse caso, peguei o restaurante com melhor polaridade

# Filtrar o DataFrame para selecionar apenas os dados do restaurante desejado
restaurante_df = freq_df[freq_df["restaurante"] == restaurante]

# Criar um dicionário com as palavras e suas frequências
palavras = dict(zip(restaurante_df["palavra"], restaurante_df["frequencia"]))

# Criar a nuvem de palavras
wordcloud = WordCloud(background_color="white").generate_from_frequencies(palavras)

# Mostrar a nuvem de palavras
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(restaurante)
plt.show()
```

[89] MagicPython

```
Digite o restaurante desejado: The Coffee
```

Conclusão de frequência de palavras

É importante lembrar que as stopwords são palavras muito comuns e genéricas que geralmente não contribuem para a análise de sentimento, mas é possível que algumas stopwords sejam importantes para a análise dependendo do contexto. Além disso, com o resultado acima, podemos analisar que a polaridade foi calculada com base de avaliações do atendimento e de comida, devido o aparecimento das palavras "atendimento"(serviço) ,"café"(comida) e "ótimo"(pode tanto ser comentário de comida quanto de serviço).

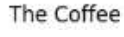
Wordcloud de polaridade e de frequência:

para cada restaurante colocado no input, sai o output da polaridade e das palavras com mais frequência

```python
# Definir o nome do restaurante que você deseja analisar
restaurante = input(str('Digite o restaurante desejado: '))

# Filtrar o DataFrame para selecionar apenas os dados do restaurante desejado
restaurante_freq_df = freq_df[freq_df["restaurante"] == restaurante]
restaurante_sent_df = sent_restaurante[sent_restaurante["restaurante"] == restaurante]

# Calcular a média da polaridade dos comentários do restaurante
polaridade_media = restaurante_sent_df["polaridade"].mean()

# Exibir legenda da polaridade
if polaridade_media > 0:
    legenda_polaridade = 'positiva'
elif polaridade_media < 0:
    legenda_polaridade = 'negativa'
else:
    legenda_polaridade = 'neutra'

print(f'A polaridade média dos comentários do restaurante {restaurante} é {polaridade_media:.2f} ({legenda_polaridade}).')

# Criar um dicionário com as palavras e suas frequências
palavras = dict(zip(restaurante_freq_df["palavra"], restaurante_freq_df["frequencia"]))
wordcloud = WordCloud(background_color='white').generate_from_frequencies(palavras)

# Mostrar a nuvem de palavras
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title(restaurante)
plt.show()
```

[98] MagicPython

```
Digite o restaurante desejado: The Coffee
A polaridade média dos comentários do restaurante The Coffee é 0.38 (positiva).
```

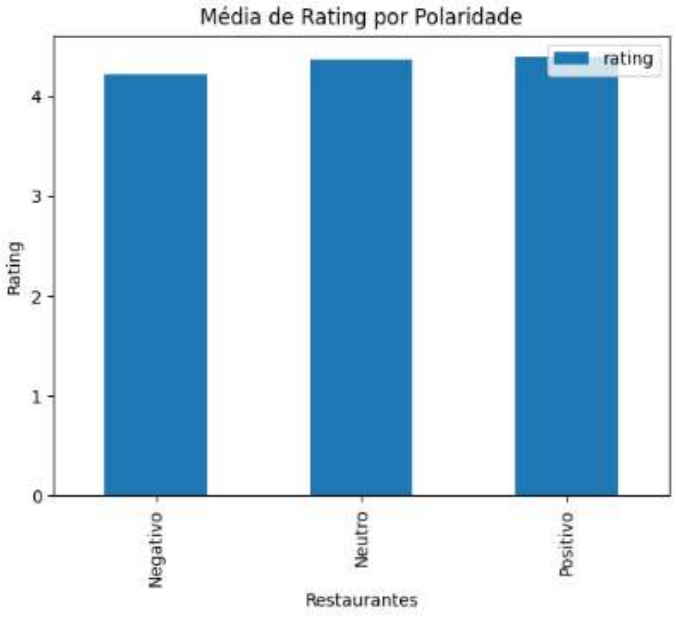


Verificando se nossa análise de polaridade corresponde com as notas dadas aos estabelecimentos

```
restaurant_sentiment = pd.read_csv('restaurant_polaridade.csv')
df = restaurant_sentiment.groupby('polaridade_texto').agg({'rating':'mean'})

# Plotar o gráfico de barras
df.plot(kind='bar')
plt.xlabel('Restaurantes')
plt.ylabel('Rating')
plt.title('Média de Rating por Polaridade')
plt.show()
```

[91] MagicPython

...



Apesar da pouca variação entre as médias, a distribuição do gráfico mostra como a nossa análise de sentimentos está coerente com a realidade, visto que, comentários classificados por nós como negativos de fato apresentam a menor média de nota de avaliação (rating), já os neutros são melhores que os negativos mas ainda ficam atrás dos positivos, como esperávamos que acontecesse.

Salvando Dados de Polaridade Média dos restaurantes

```
def summarize_text(text):
    # Transforma o texto em uma lista de frases
    sentences = text.split('. ')
    # Cria um objeto parser
    parser = PlaintextParser.from_string(text, Tokenizer("english"))
    # Gera o resumo selecionando as frases mais relevantes
    summarizer = LsaSummarizer()
    summarizer.stop_words = [' ']
    summary = summarizer(document=parser.document, sentences_count=2)

    # Retorna o resumo e as pontuações das frases
    return ' '.join(map(str, summary))
```

[92] MagicPython

```
restaurant_polaridade_media = sentimento_df.groupby(['restaurante'], as_index=False).agg({'polaridade': 'mean', 'rating': 'mean'})
sentimento_df.fillna('', inplace=True)
restaurant_polaridade_media_comentarios = sentimento_df.groupby(['restaurante'], as_index=False)['comentario'].apply('. '.join).reset_index()
restaurant_polaridade_media_comentarios['ResumoTexto'] = restaurant_polaridade_media_comentarios['comentario'].apply(summarize_text)
```

[93] MagicPython

```
/usr/local/lib/python3.9/dist-packages/sumy/summarizers/lsa.py:76: UserWarning: Number of words (5) is lower than number of sentences (7). LSA algorithm may
  warn(message % (words_count, sentences_count))
/usr/local/lib/python3.9/dist-packages/sumy/summarizers/lsa.py:76: UserWarning: Number of words (2) is lower than number of sentences (8). LSA algorithm may
  warn(message % (words_count, sentences_count))
/usr/local/lib/python3.9/dist-packages/sumy/summarizers/lsa.py:76: UserWarning: Number of words (3) is lower than number of sentences (7). LSA algorithm may
  warn(message % (words_count, sentences_count))
/usr/local/lib/python3.9/dist-packages/sumy/summarizers/lsa.py:76: UserWarning: Number of words (2) is lower than number of sentences (9). LSA algorithm may
  warn(message % (words_count, sentences_count))
/usr/local/lib/python3.9/dist-packages/sumy/summarizers/lsa.py:76: UserWarning: Number of words (5) is lower than number of sentences (8). LSA algorithm may
  warn(message % (words_count, sentences_count))
/usr/local/lib/python3.9/dist-packages/sumy/summarizers/lsa.py:76: UserWarning: Number of words (4) is lower than number of sentences (7). LSA algorithm may
  warn(message % (words_count, sentences_count))
```

```
restaurant_polaridade_media_comentarios = restaurant_polaridade_media_comentarios[['ResumoTexto', 'restaurante']]
restaurant_polaridade_media = restaurant_polaridade_media.merge(restaurant_polaridade_media_comentarios, how='left', on ='restaurante')
restaurant_polaridade_media[['polaridade_num', 'polaridade_texto']] = restaurant_polaridade_media['polaridade'].apply(polaridade).apply(pd.Series)
```

[94] MagicPython

```
df_rest = pd.read_csv('most_relevant_reviews.csv',sep = ',', usecols=['title', 'place_id']).drop_duplicates()

restaurant_polaridade_media = restaurant_polaridade_media.merge(df_rest, right_on='title', left_on='restaurante', how='left')
restaurant_polaridade_media = restaurant_polaridade_media[['place_id', 'title', 'rating', 'polaridade', 'polaridade_num', 'polaridade_texto', 'ResumoTexto
restaurant_polaridade_media.to_csv("restaurant_polaridade_media.csv", index=False)
```

[95] MagicPython

Dashboard

O nosso principal objetivo com o dashboard é **aprimorar e otimizar as sugestões fornecidas pela plataforma do Google** (API's do Google Maps Reviews e Google Maps Place Results) para os usuários. Por meio do dashboard, o usuário pode realizar buscas específicas e utilizar todos os filtros desejados em um *ambiente visual dinâmico*.

O primeiro dashboard é direcionado para **buscas gerais, em que o usuário pode procurar por estabelecimentos com base em tópicos de sua preferência**, como

O primeiro dashboard é direcionado para **buscas gerais, em que o usuário pode procurar por estabelecimentos com base em tópicos de sua preferência**, como zona, quantidade de avaliações, categoria e tipo de serviço. A partir dessas escolhas, são exibidas informações relevantes sobre os estabelecimentos, como a quantidade total, a média de nota e de preço.

O segundo dashboard é direcionado para ,**usuários que já sabem o local que desejam visitar**. Nesse caso, basta digitar o nome do estabelecimento para obter **detalhes precisos**, como avaliações positivas e negativas dos clientes, as palavras mais utilizadas nos comentários e informações importantes, como horários de funcionamento, categoria, zona e endereço no mapa. Além disso, são fornecidas sugestões de locais semelhantes ao pesquisado.

O grande diferencial do nosso dashboard é a **otimização do tempo do usuário de busca**. Todos os recursos foram projetados para que o usuário encontre rapidamente o que procura e possa tomar decisões informadas. O ambiente visual dinâmico, os filtros e as informações detalhadas fornecidas pelos dashboards permitem que o usuário *economize tempo e encontre facilmente o estabelecimento ideal.*

Neste link você pode encontrar a versão digital do nosso Dashboard para explorar

Para que o usuário possa ter maior facilidade com o dashboard aqui trazemos um exemplo de busca aonde o usuário está procurando informações detalhadas sobre um restaurante chamado "A Creperia". Utilizando o segundo dashboard de estabelecimentos específicos, basta digitar o nome do estabelecimento na barra de pesquisa. Em seguida, serão exibidas informações detalhadas sobre o local, incluindo avaliações de clientes, horários de funcionamento, dias da semana em que o estabelecimento está aberto, a categoria a que pertence e o endereço no mapa.

É interessante observar esse caso, qual podemos notar que existe algumas avaliações que apesar de ter comentários falando bem sobre o estabelecimento, ou seja um comentário positivo, a sua avaliação é de uma estrela, isso provavelmente ocorreu porque a pessoa pode ter pensado que 1 estrela é a melhor avaliação para o restaurante, sendo assim notamos como é relevante termos um modelo que analise aém das estrelas e que olhe para as palavras nos comentários de clientes.

EXPLICAÇÃO DAS TÉCNICAS DE IA APLICADAS PARA FAZER PREDIÇÃO DE DADOS E SUA RELAÇÃO COM OS OBJETIVOS

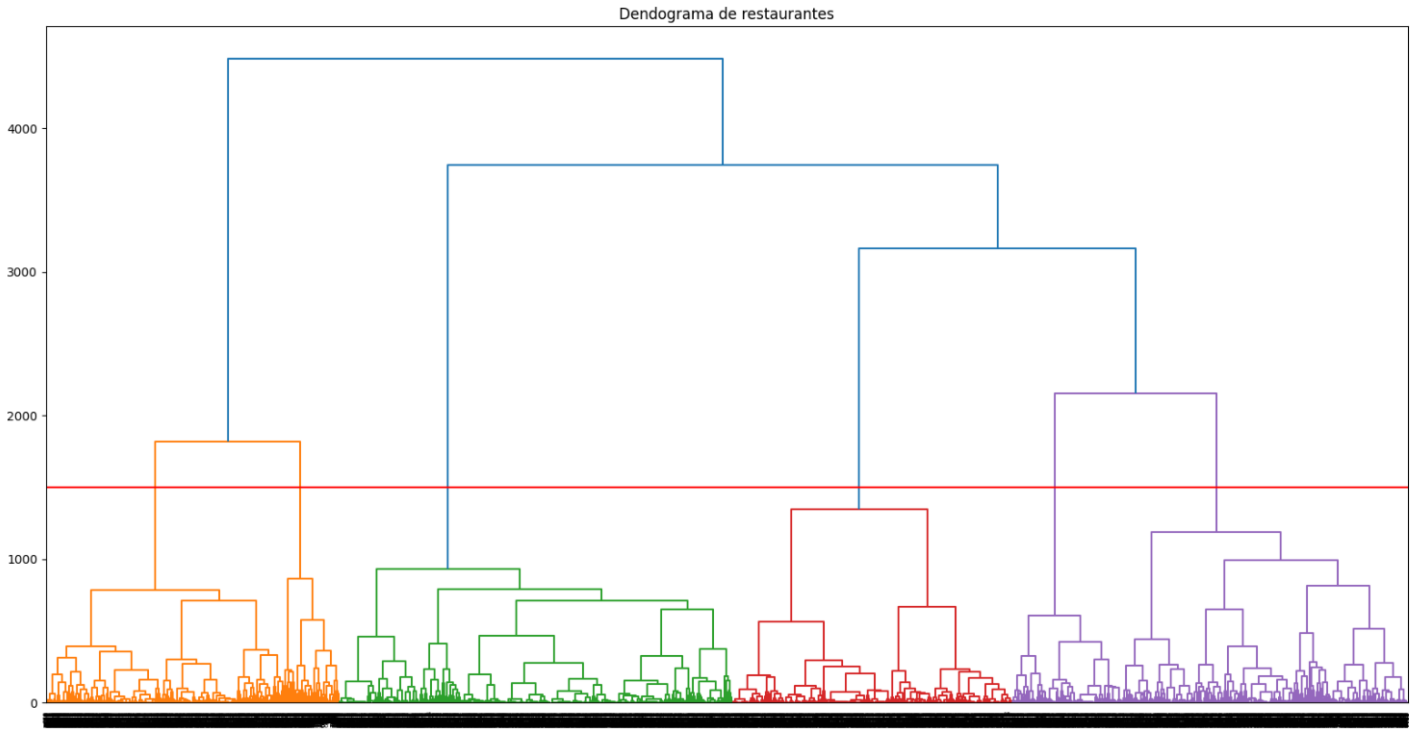
O **modelo K-Means clustering** é uma técnica popular de agrupamento que pode ser usada para segmentar estabelecimentos de acordo com suas informações gerais. O algoritmo funciona agrupando as observações em clusters ou grupos, com base na similaridade de suas características. No contexto de **segmentação de estabelecimentos**, essas características podem incluir dados de funcionamento, como dias, horários e lotação, bem como informações sobre a localização geográfica dos estabelecimentos e outros dados relevantes.

Ao aplicar o modelo K-Means clustering a um conjunto de dados de estabelecimentos, o **algoritmo tenta encontrar grupos de estabelecimentos que compartilham características semelhantes**. Esses grupos podem ser usados para entender melhor as necessidades e preferências dos diferentes segmentos de clientes e estabelecimentos.

Após a segmentação dos estabelecimentos em clusters, é possível usar esses grupos para fazer previsões e tomar decisões mais precisas. Por exemplo, diversos usuários podem usufruir da clusterização para encontrarem estabelecimentos similares, além de que ainda se pode ter uma análise mais profunda em tipos de estabelecimentos por cluster, para pessoas e para pesquisa de mercado quando colocamos os dados disponíveis para o público de forma facilitada e organizada.

É importante ressaltar que o dendograma é um gráfico utilizado para analisar a relação entre os clusters de dados, nas quais clusters mais distantes e com alturas maiores apresentam maior dissimilaridade, mostrando, muitas vezes, uma boa segmentação.

Entretanto, como no caso do nosso trabalho, muitas vezes é importante que a divisão seja feita em clusters menores para que a qualidade da segmentação seja elevada. No projeto em questão, foi possível avaliar que o número ideal de clusters era 16, mesmo que não apresentassem as maiores distâncias, isso porque grupos visualmente próximos muitas vezes apresentavam diferenças importantes levando em decorrência da similaridade dos dados analisados.

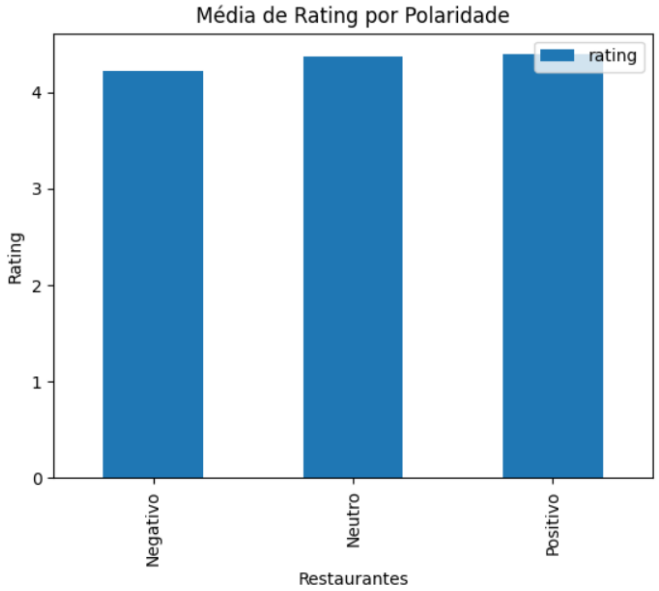


COMO AVALIAR ESTE TRABALHO CONSIDERANDO NOSSOS OBJETIVOS

Levando em consideração os objetivos do trabalho, para avaliar o resultado final é importante considerar se o mesmo:

- **Está fornecendo ao usuário uma lista de estabelecimentos com informações relevantes coletadas a partir do consumo de dados da API serpAPI.**
    - Isso pode ser observado através dos dados fornecidos tanto no dashboard quanto API, que são valiosos tanto para o julgamento dos usuários sobre os estabelecimentos quanto para pesquisas de mercado.

- **Fornece dados concisos de reviews de usuários, como resumo de comentários e análise de sentimentos, obtidos através das técnicas de NPL e que estejam classificados de forma coerente com a realidade.**
    - Isso pode ser observado ao comparar a média das notas de avaliações dadas pelos usuários para cada grupo de polaridade - polaridade positiva tem média maior que polaridade neutra que, por sua vez, tem média maior que polaridade negativa. Além disso, em alguns casos, a análise de sentimentos se mostrou mais precisa que as notas dadas pelos usuários como exemplificado no nosso caderno Jupyter, na parte de Dashboard.



- **Apresenta sugestão de estabelecimentos parecidos levando em conta as informações utilizadas na modelagem.**
    - Pode-se observar isso de acordo com o dendograma, apresentado na parte de explicações das técnicas de IA aplicadas para fazer predição de dados e sua relação com os objetivos deste PDF, que mostra uma distância satisfatória entre os clusters dentre os quais os restaurantes foram distribuídos.

- **Tem Documentação clara das etapas necessárias para o resultado final como a análise de dados, modelagens, processamentos, construção da base de dados e API.**
    - A documentação está disponível no presente documento e no caderno Jupyter com um bom nível de detalhamento, de forma que todos os processos podem ser reproduzidos.

- **Fornece o produto final de forma acessível através de um dashboard e API funcionais e acessíveis.**
    - Os exemplos de uso, em nosso caderno Jupyter, de ambos os serviços mostram a eficiência e acessibilidade de ambos.

PRÓXIMOS PASSOS

O projeto é uma excelente versão inicial de um produto com grande potencial de valor para o mercado. Atualmente, a internet desempenha um papel importante na visibilidade dos estabelecimentos, sendo as avaliações um pilar relevante, especialmente nas redes sociais como o Instagram e o TikTok, as quais possuem um nicho bastante relevante referente a reviews de locais, que movimenta um volume expressivo de dinheiro e gera muitas visualizações e público engajado.

Isso é reflexo do grande interesse das pessoas em conhecer mais sobre a diversidade de opções e propostas e, por isso, ter acesso a informações precisas e confiáveis sobre os estabelecimentos tornou- se uma questão fundamental. Nesse sentido, nosso projeto oferece de maneira acessível uma ampla variedade de informações sobre os locais, incluindo sugestões de novos lugares que possam ser do interesse do usuário.

Para melhorar esse produto, os próximos passos seriam:

- ☐ Coletar um maior número de dados, sobretudo de avaliações;
- ☐ Tornar o modelo de segmentação mais poderoso, levando em conta informações do usuário e não apenas da similaridade entre as características dos estabelecimentos;
- ☐ Replicar as funcionalidades do Dashboard em uma interface ainda mais acessível para o público geral e não apenas para aqueles que utilizam Power BI;
- ☐ Exploração de parcerias com estabelecimentos e influenciadores a fim de promover a utilização do produto e expandir sua base de usuários;
- ☐ Além disso, pode-se oferecer serviços adicionais, como a criação de perfis de estabelecimentos, usuários, influenciadores e campanhas de marketing personalizadas.