



# Agrupando Bairros de Porto Alegre

---

VINICIUS BRANCO SCORTEGAGNA

02/05/2022

# *Problema Proposto*

# Introdução

---

- ❑ Para um empreendedor que deseja investir em Porto Alegre, é conveniente investigar as características de cada bairro da cidade e agrupá-los de acordo com semelhanças e diferenças em relação aos diversos tipos de serviços oferecidos.
- ❑ Com isso, é possível então decidir em qual ramo de negócios será feito o investimento.
- ❑ Um modelo de *clusterização*, portanto, pode produzir valiosos subsídios à tomada de decisão estratégica.

# *Dados Utilizados*

COLETA E TRATAMENTO DE DADOS

# Webscraping das informações sobre bairros de Porto Alegre

---

- ❑ As informações sobre bairros de Porto Alegre foram encontradas na Wikipédia: [Lista de bairros de Porto Alegre](#). O web scraping neste caso pôde ser feito de duas maneiras:
  - ❑ Entrar no site e acessar o item “inspecionar” do menu do botão direito, encontrar o trecho do HTML correspondente à tabela, copiar o texto que se encontra entre `<table>` e `</table>` e colar em um arquivo de texto com final `.html` para ser lido pelo comando `read_html(caminho)` da biblioteca `pandas`.
  - ❑ Utilizando o método `.get(url)` da biblioteca `requests` com a URL da página supracitada e o objeto `BeautifulSoup()` da biblioteca `bs4` para interpretar o HTML da página inteira, filtrando tabelas, linhas e conteúdos das células com o comando `.find_all(...)` e organizando em um `dataframe` através de um laço.

# Obtenção das coordenadas centrais de cada bairro

---

- ❑ Foram utilizadas as bibliotecas geocoder e geopy de forma mesclada, de acordo com os resultados de cada uma.
- ❑ Com o geopy, foi acessado o [Open Street Map](#) (Nominatim);
- ❑ Com o geocoder, foi utilizado o método .arcgis().
- ❑ Graças ao web scraping que fora realizado, foram obtidas as áreas da maioria dos bairros, permitindo basear-se nelas para estabelecer um raio de busca para cada bairro, a fim de utilizar em posterior requisição à API do Foursquare.

# Obtenção das coordenadas centrais de cada bairro

---

```
1 def latlon(bairro):
2     coord = None
3     while(coord is None):
4         g = geocoder.arcgis(f'{bairro}, Porto Alegre, Brasil')
5         coord = g.latlng
6     return coord
```

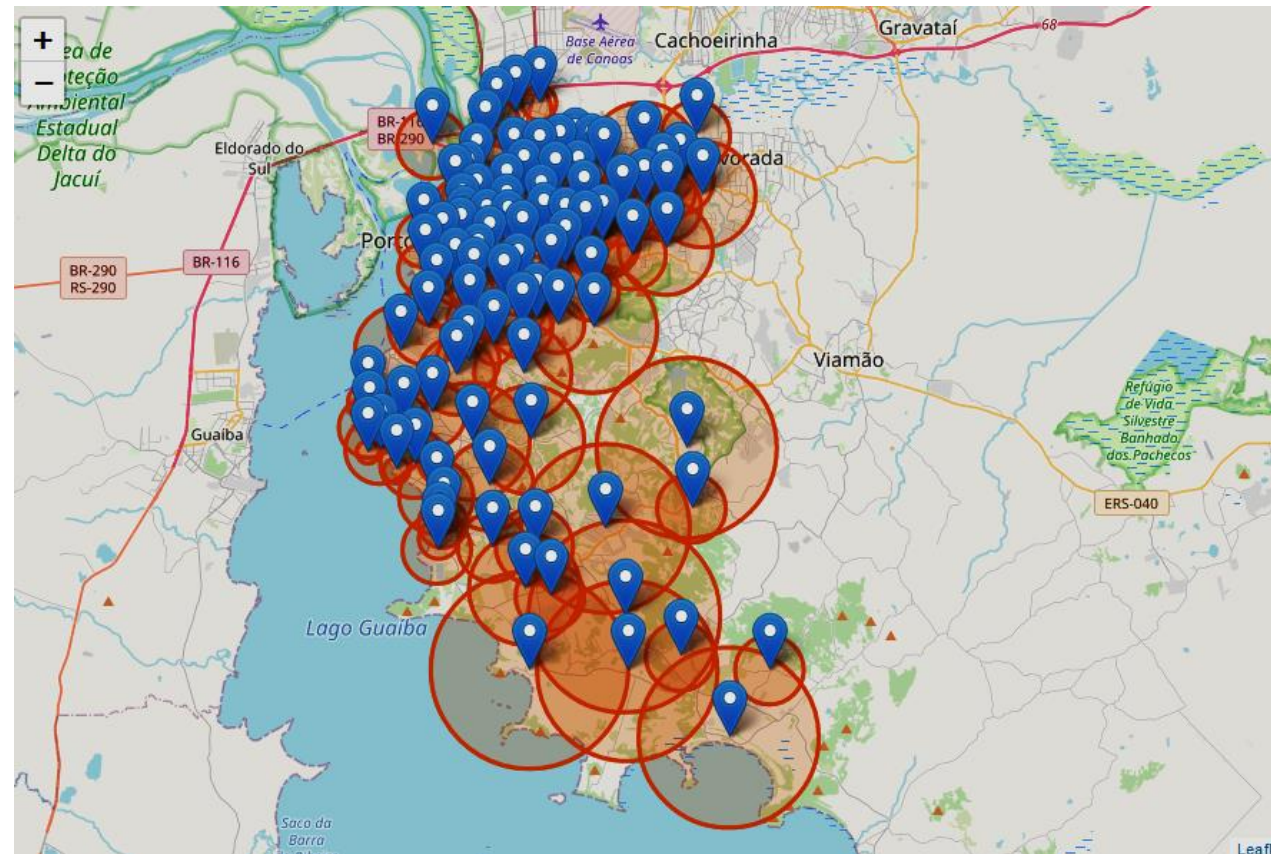
```
1 geolocator = Nominatim(user_agent='TestBot')
2 latitudes = []
3 longitudes = []
4 errados = [
5     'Camaquã',
6     'Cristal',
7     'Menino Deus',
8     'Morro Santana',
9     'Santo Antônio',
10    'Vila Ipiranga'
11 ]
```

```
12 for bairro in lista_bairros:
13     if bairro in errados:
14         g = latlon(bairro)
15         latitudes.append(g[0])
16         longitudes.append(g[1])
17     else:
18         g = geolocator.geocode(f"{bairro}, Porto Alegre, Brasil")
19         latitudes.append(g.latitude)
20         longitudes.append(g.longitude)
21 df['lat'] = latitudes
22 df['lon'] = longitudes
23 # df.to_csv(caminho + 'bairros_geoloc_3.csv')
```



# Visualizando no mapa com Folium

---





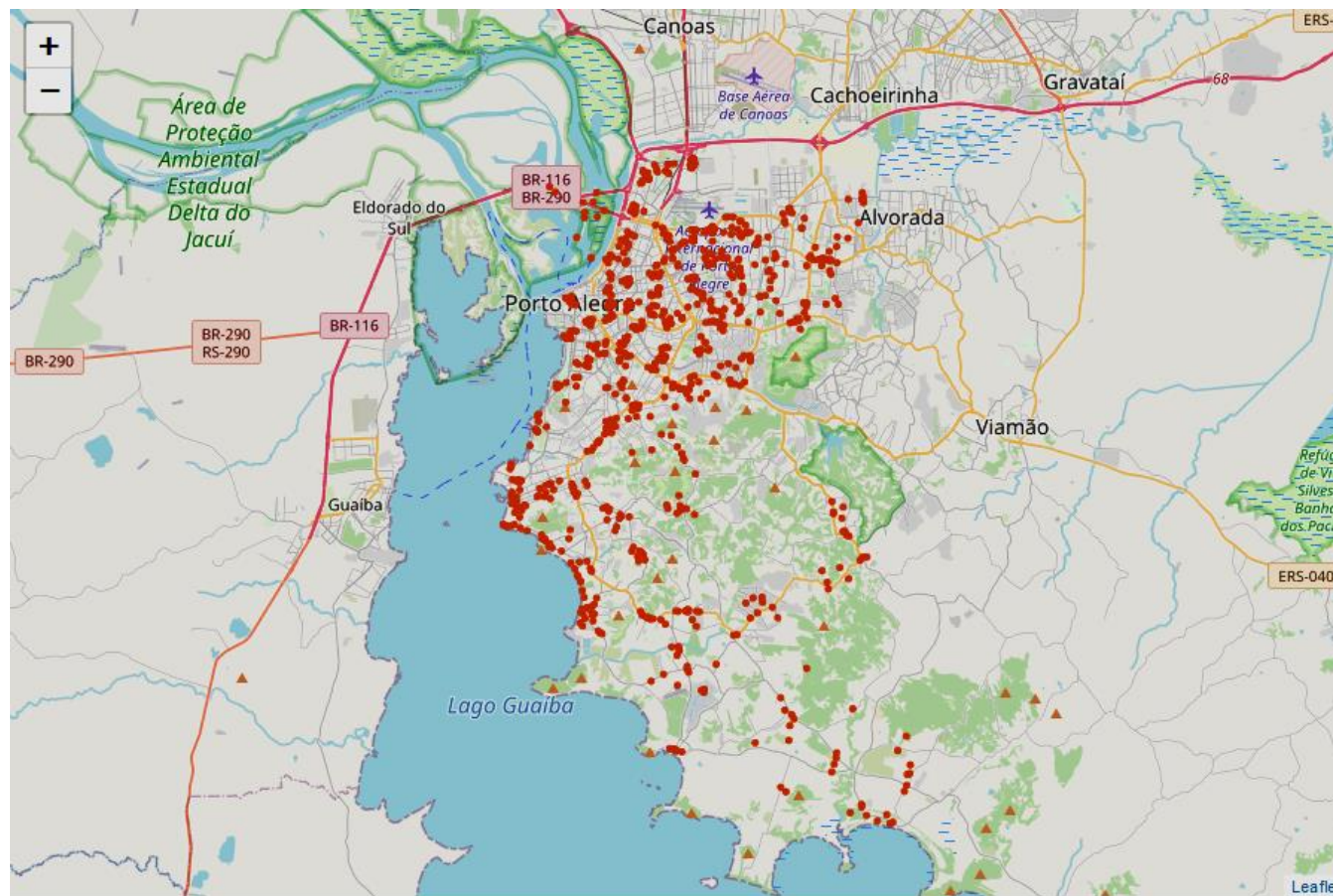
# API do Foursquare para dados de localidades

---

- ❑ A API do Foursquare foi utilizada para obter uma amostra de locais para cada centróide.
- ❑ As linhas duplicadas são eliminadas, mantendo apenas linhas únicas com base no código identificador de cada localidade.

# Coordenadas dos locais obtidos

---



# Determinação do bairro de cada localidade

---

- ❑ A maior parte dos locais obtidos nas requisições ao Foursquare está faltando.
- ❑ Os bairros de cada localidade, contudo, podem ser recuperados com o geopy, utilizando o método `.reverse()`, no qual se fornecem as coordenadas e informações de endereço associadas ao ponto são retornadas.

# *Métodos Utilizados*

EXPLORAÇÃO DE DADOS

MODELO DE CLUSTERIZAÇÃO

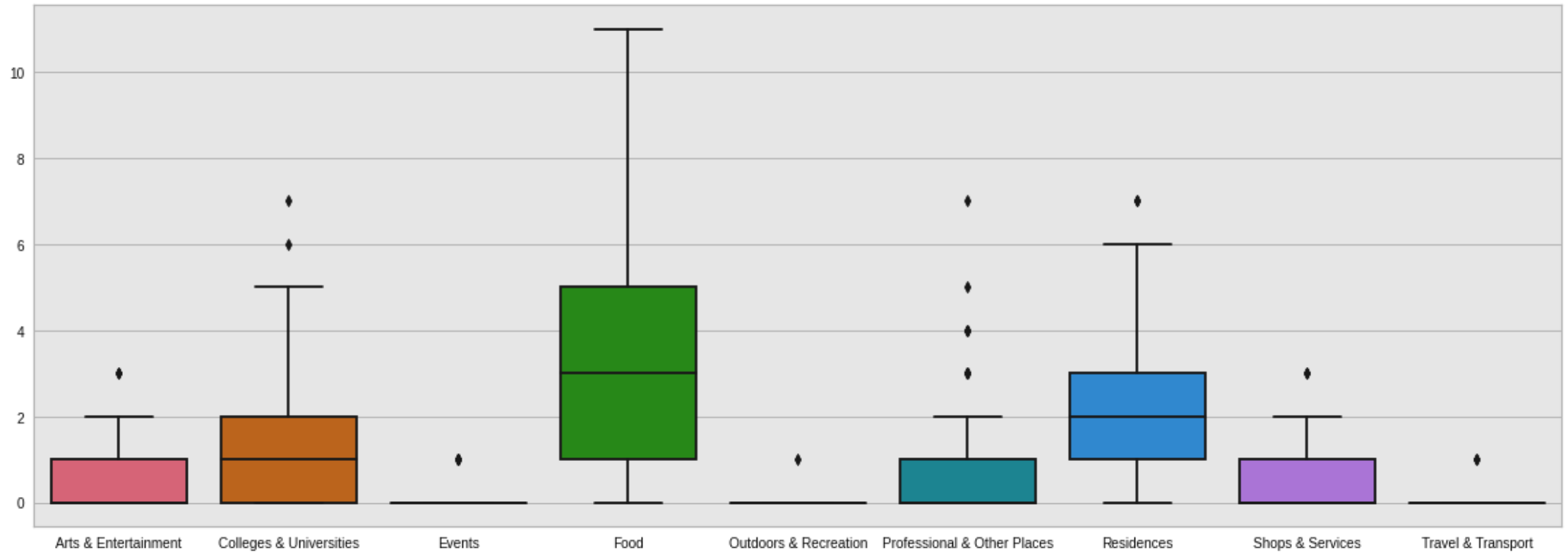
# Categorias do Foursquare

---

categoria
Arts & Entertainment
Colleges & Universities
Events
Food
Nightlife Spots
Outdoors & Recreation
Professional & Other Places
Residences
Shops & Services
Travel & Transport

# Boxplots de cada categoria

---



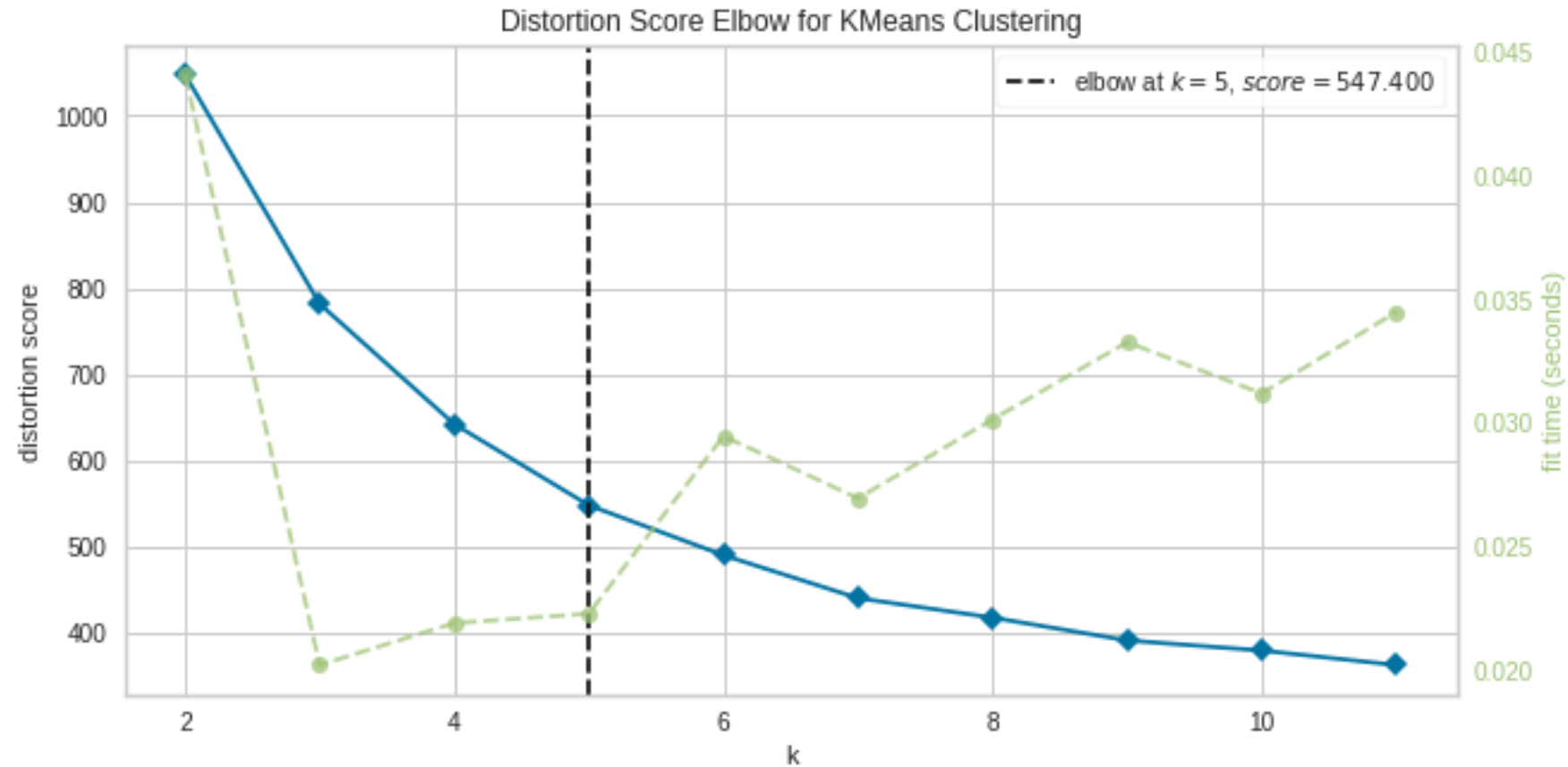


# Modelo KMeans

---

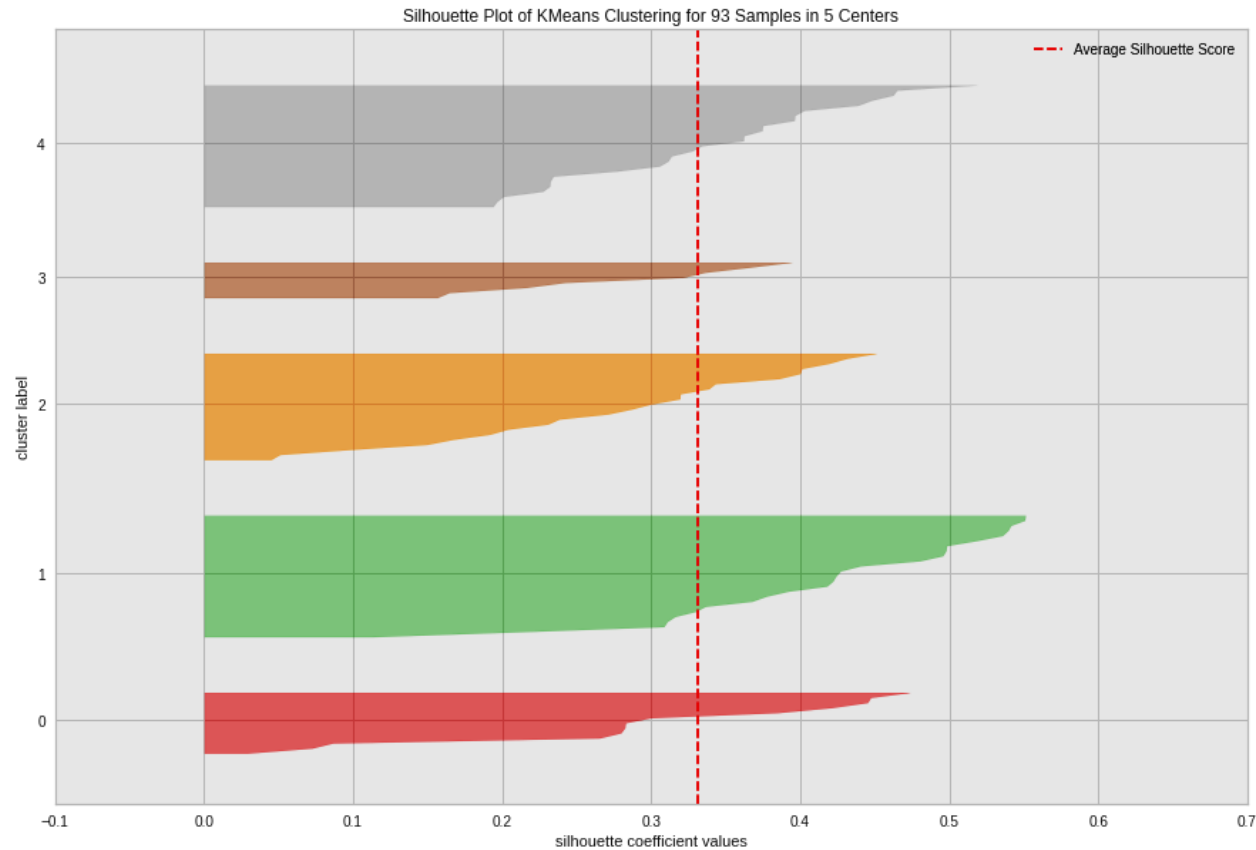
- ❑ As quantidades de locais de cada categoria do Foursquare para cada bairro foram agrupados num dataframe para clusterização.
- ❑ O modelo KMeans foi aplicado com auxílio do Distortion Score Elbow e do Silhouette Plot.
- ❑ Foi utilizado `random_state=42` para assegurar a reprodutibilidade.

# Elbow Method indicando 5 grupos



# Silhouette Plot com 5 clusters

---



# *Resultados Obtidos*

CLUSTERS

# Bairros Clusterizados

Legenda = {

0: 'blue',

1: 'green',

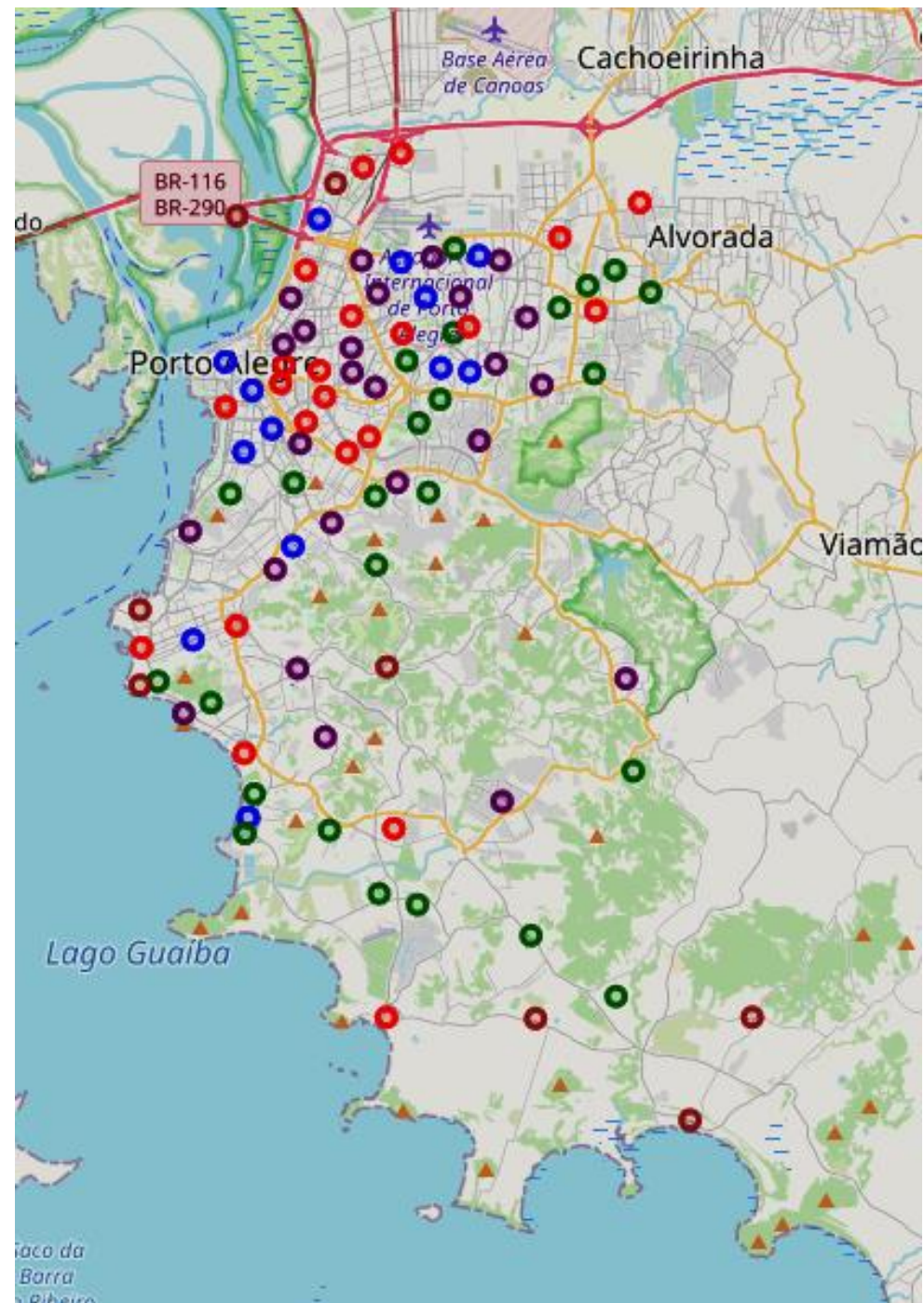
2: 'red',

3: 'brown',

4: 'purple'

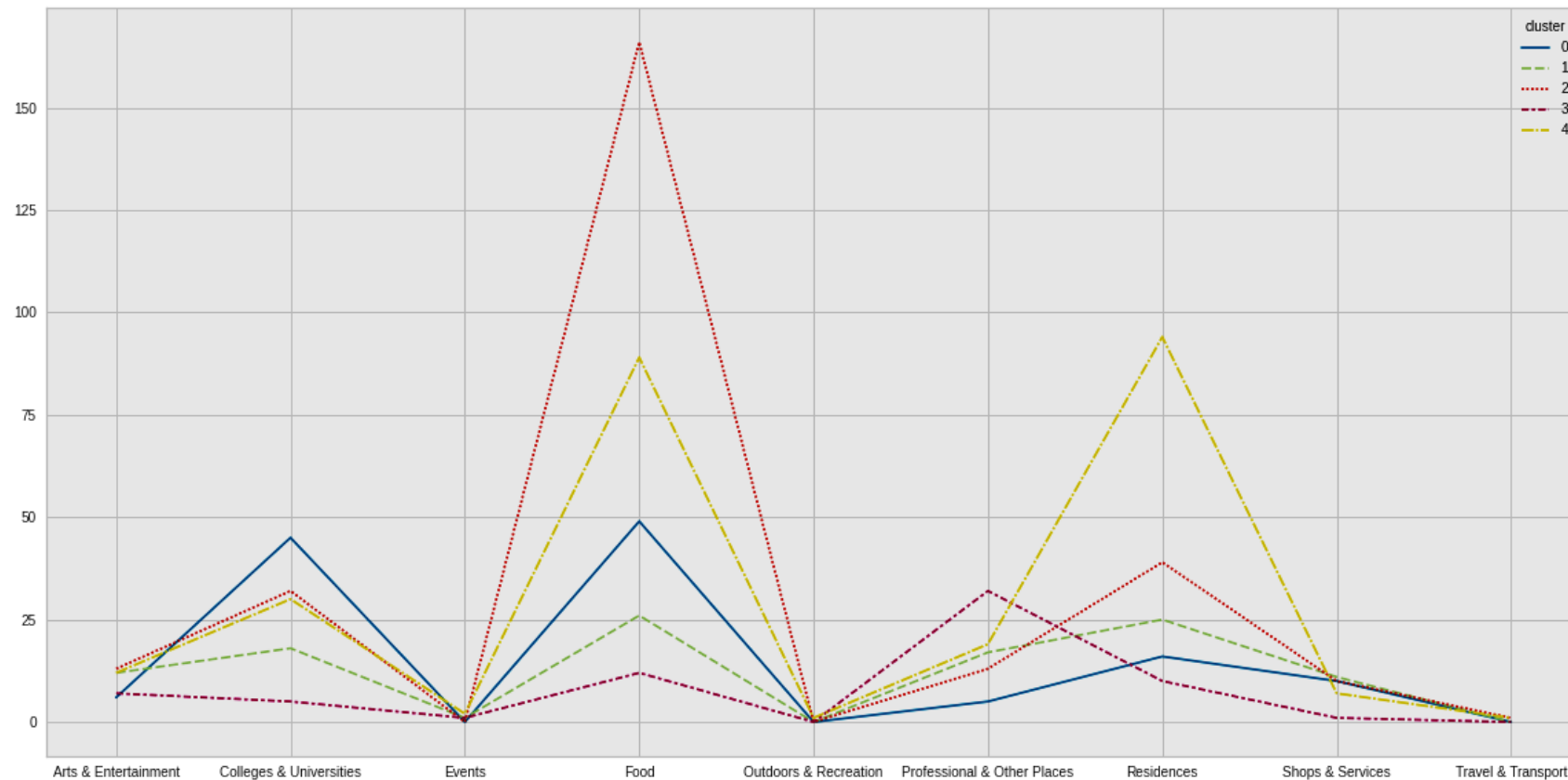
}

02/05/2022



# Número total de localidades de cada categoria por cluster

---





# *Discussão e Conclusão*

# Características dos clusters

---

- ❑ O cluster número 2 possui mais estabelecimentos do setor de alimentação e menos residências;
- ❑ O cluster número 4 possui um equilíbrio entre essas duas categorias;

# Conclusão

---

- ❑ A baixa quantidade de amostras impede que se atinjam conclusões analíticas robustas;
- ❑ Isso enseja a continuidade do trabalho, efetuando-se uma ampliação das buscas com a API do Foursquare ou em outras fontes de dados.

*Obrigado*