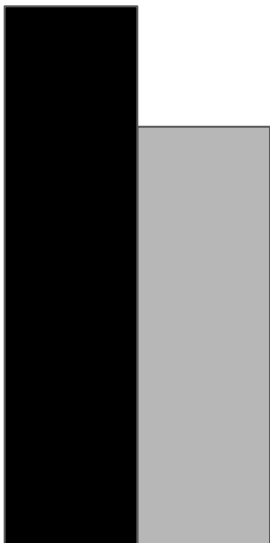
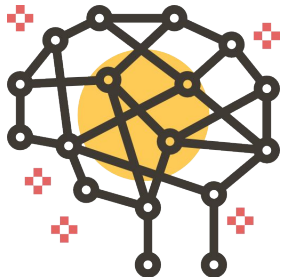


# **Inteligência Artificial - Aprendizado de Máquina**



Guilherme de Lima - 201910055811

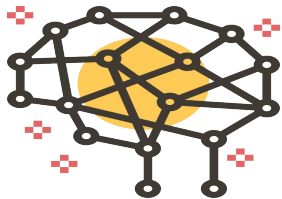


# Estudo de Caso



**O atual prefeito Eduardo Paes obrigou as empresas de ônibus a circular em 100% de seus efetivos de ônibus com ar-condicionado , porém quem utiliza esse meio de transporte repara que a maioria dos efetivos não possuem. Por isso ficam alguns questionamentos:**

- **A maioria dos ônibus serão descartados e algumas empresas irão rodar com um número menor de ônibus?**
- **Com o ar-condicionado e o preço do combustível, algumas empresas terão prejuízo devido a atual rota de ônibus? As atuais rotas são as melhores?**



# DataSet



## Pontos de Parada da rede de transporte público por ônibus (SPPO)



**Publicador Prefeitura**

Prefeitura da Cidade do Rio de Janeiro

### Resumo

A camada contém os pontos de parada da rede de transporte público por ônibus (SPPO) contemplados no GTFS mais recente.

[Visualizar Detalhes Completos](#)

[Download](#)

### Detalhes



**Conjunto de Dados**

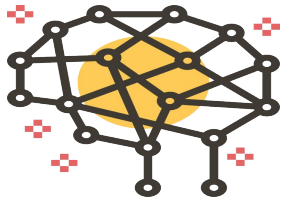
Feature Layer



**1 de setembro de 2023**

Informação Atualizada

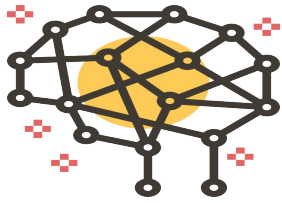




# Atributos



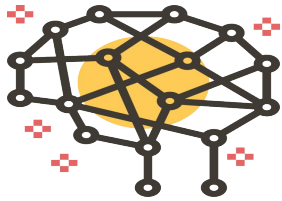
- **X:** coordenada x
- **Y:** coordenada y
- **fid:** id
- **stop\_name:** nome do ponto



# Pré-Processamento (Redução)



```
dados = dados.drop('fid', axis = 1)
```



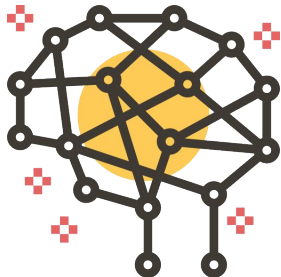
# Pré-Processamento

## (Verificação de dados faltantes)



```
dados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7398 entries, 0 to 7397
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype  
---  -
0   X                7398 non-null  float64
1   Y                7398 non-null  float64
2   stop_name       7398 non-null  object 
dtypes: float64(2), object(1)
memory usage: 173.5+ KB
```

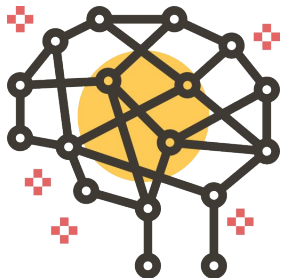


# K-Means

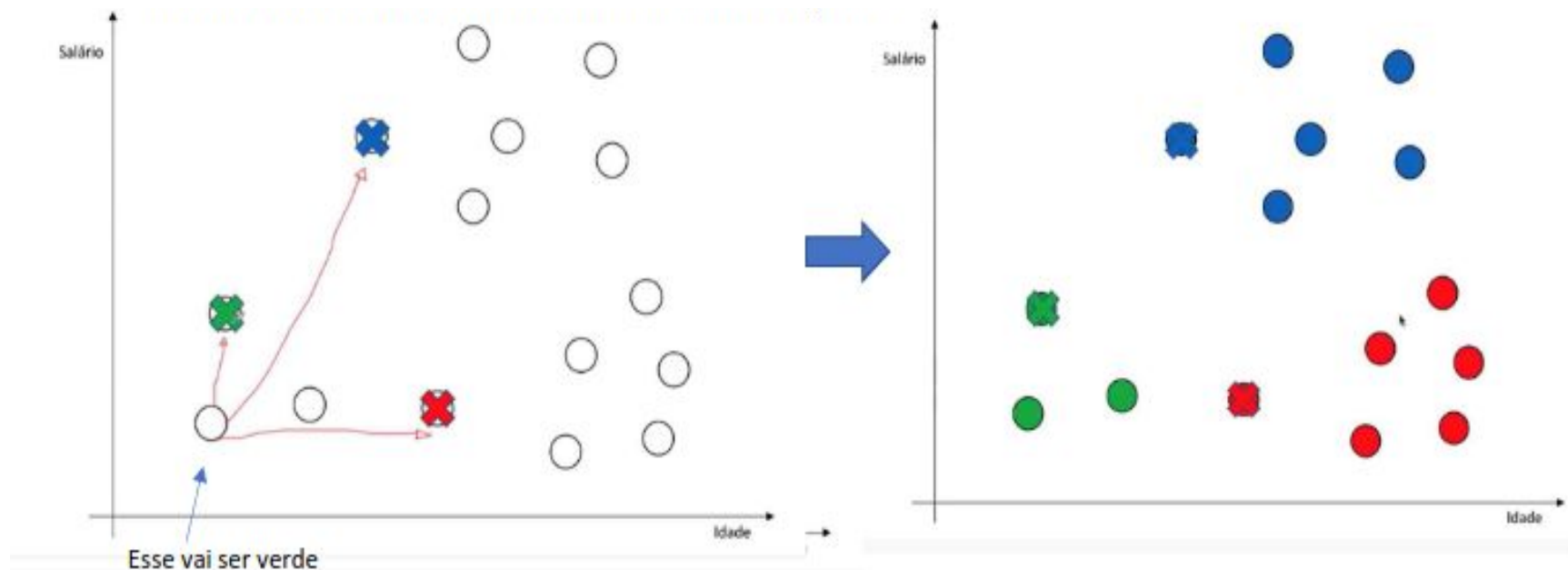


**Algoritmo sem supervisão que fornece uma solução para o problema de agrupamento. O algoritmo segue um procedimento para formar clusters que contêm pontos de dados homogêneos**

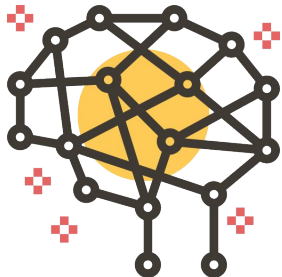
- **De acordo com o  $k$  de entrada e selecionados  $k$  centróides, conectando os pontos vizinhos e criando um cluster. Repetindo o processo até que os centróides não mude**
- **O  $k$  de entrada será considerado nosso número de rotas**



# K-Means







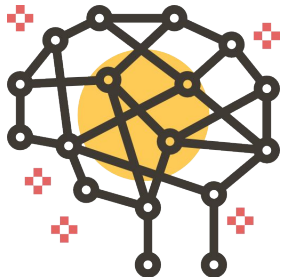
# Implementação



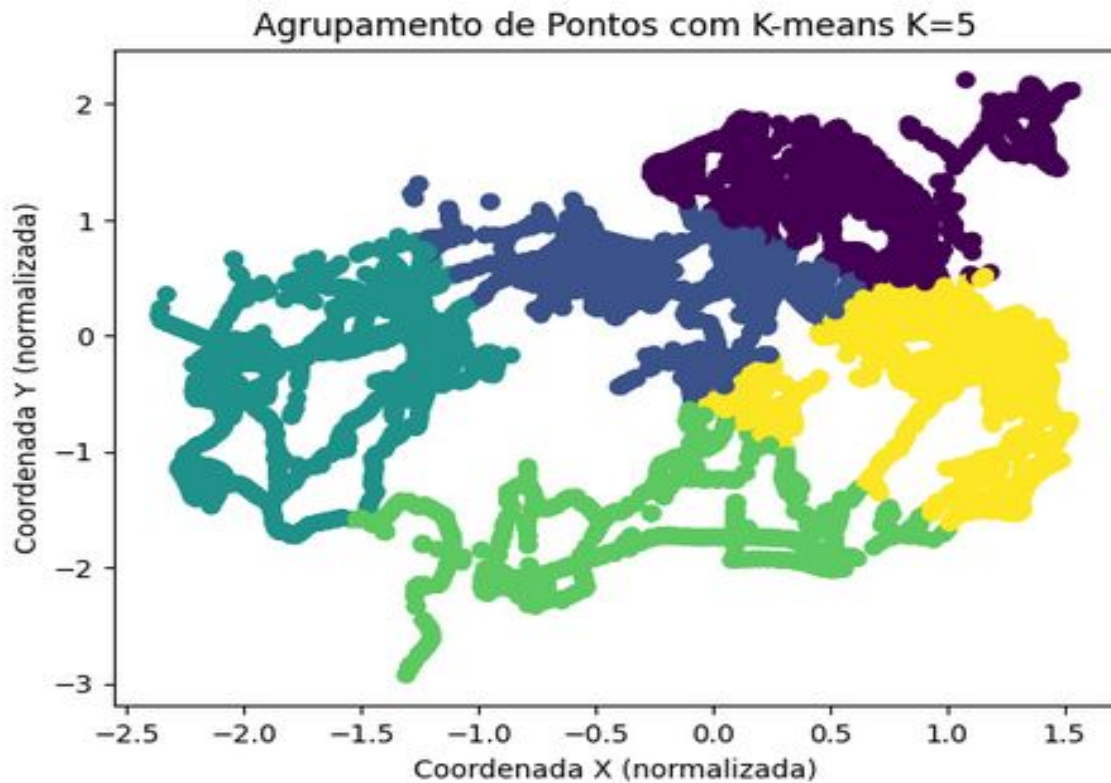
**Escolhemos a biblioteca scikit-learn em python para implementar o k-means, onde selecionarem o k(rotas) de entrada para ser nosso número de clusters:**

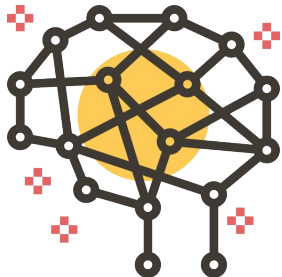
**(kmeans = KMeans(n\_clusters=k, random\_state=42))**

**Após os resultado faremos a integração com a biblioteca matplotlib.pyplot para mostrar os resultados.**

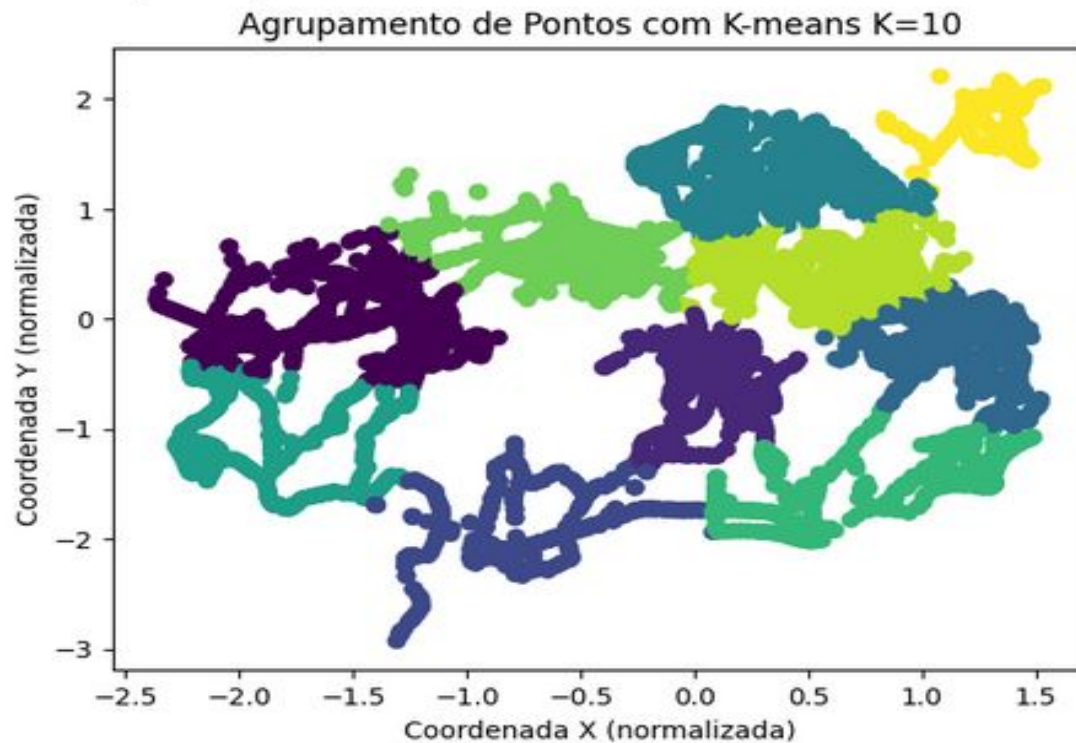


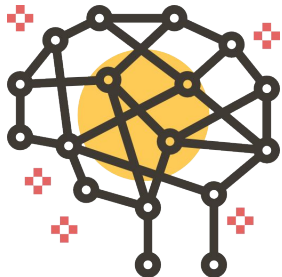
# 5 ROTAS



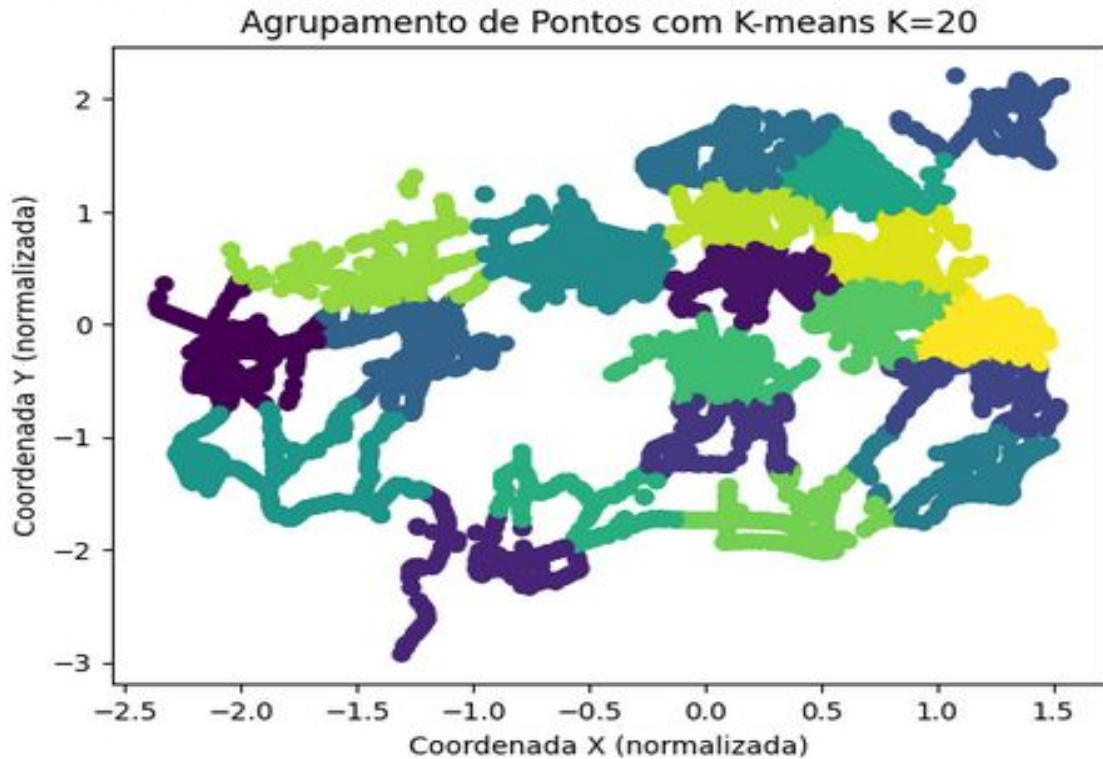


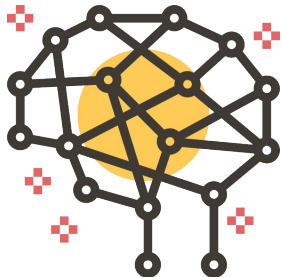
# 10 ROTAS



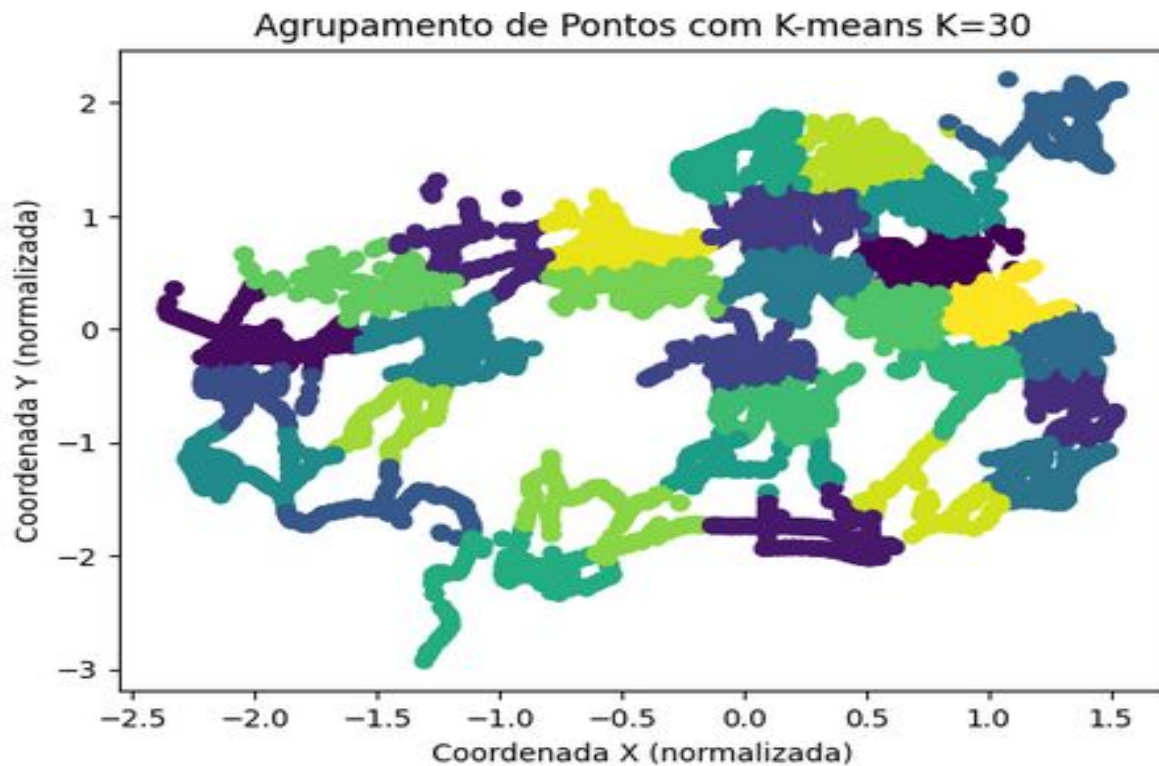


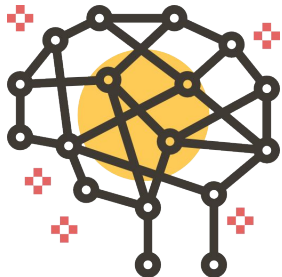
# 20 ROTAS



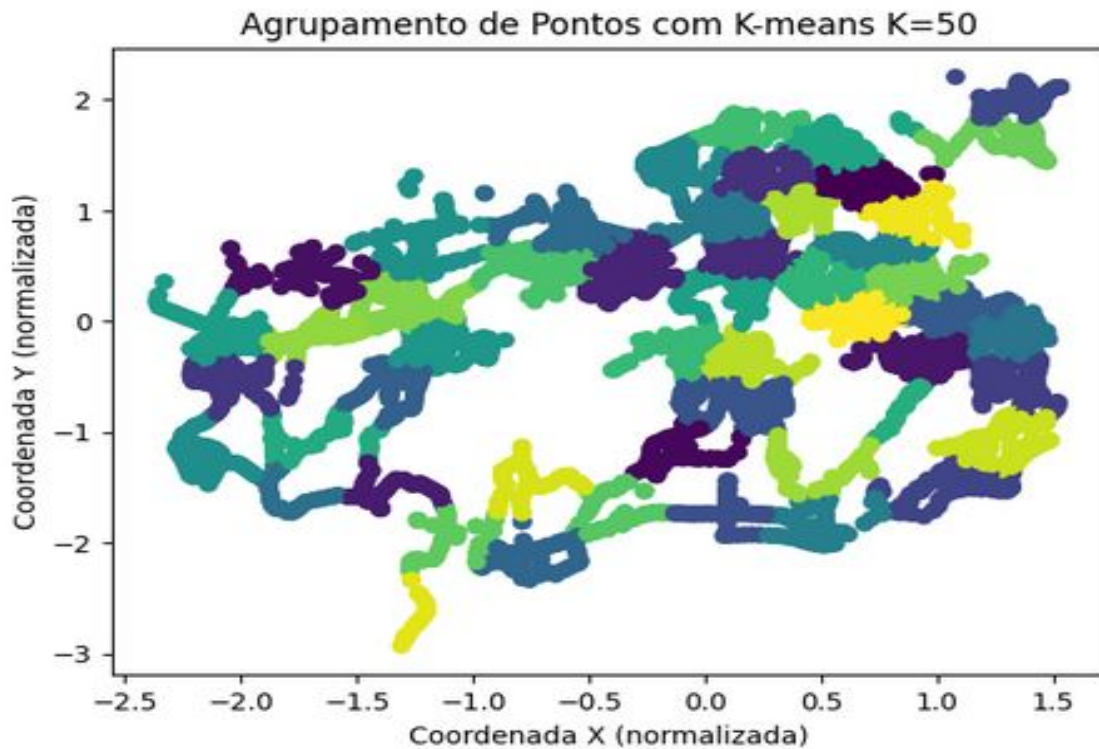


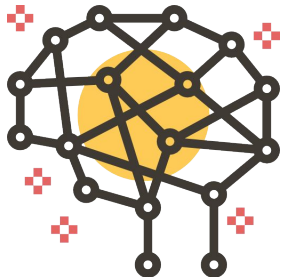
# 30 ROTAS





# 50 ROTAS





# Conclusões



- Quanto maior o número de rotas, maior a probabilidade da população ter que utilizar a baldeação para chegar no seu destino, porém os ônibus economizarão em gasto e quilometragem . Sendo necessário uma mudança no sistema de cobrança
  - Rotas em menor número além de uma distância maior as empresas irão gostar com gasolina
  - Após definirmos as rotas podemos reformular o itinerário de linhas já existentes também com a ajuda de aprendizado de maquina com o dataset existente ([https://datariov2-pcrj.hub.arcgis.com/datasets/2b08dfadd6e94045a925df244efcea7e\\_1/explore](https://datariov2-pcrj.hub.arcgis.com/datasets/2b08dfadd6e94045a925df244efcea7e_1/explore))