

Giovana Vieira de Moraes 587591  
Ingrid Maria Santos Pires 620300  
Lucas Rigo Yoshimura 620092  
Matheus da Silva Jesus 620025

## **Relatório de Inteligência Artificial**

### **Introdução**

Este projeto tem como objetivo a implementação de três algoritmos - k-médias, single-link e average-link - os quais foram implementados em linguagem de programação C e Python. Geramos as partições resultantes a partir de 3 datasets fornecidos pela professora e temos aqui o intuito de discutir esses resultados, o desempenho dos algoritmos e a diferença de resposta para cada configuração de clusters.

### **Chamada de Algoritmos**

#### **K-médias**

```
python kmeans.py <nome_dataset> <k> <numIteracoes>
```

#### **Single-link**

```
python single_link.py <nome_dataset> <kMin> <kMax>
```

#### **Average-link**

```
./avg <nome_dataset> <kMin> <kMax>
```

#### **Índice Rand**

```
python ars.py <resultadoReal> <resultadoAlgoritmo>
```

### **Desempenho**

- **K-médias**

- **c2ds1-2sp.txt**

- com k = 2 e 10 iterações o tempo de processamento foi de 0.0972 segundos
    - com k = 3 e 10 iterações o tempo de processamento foi de 0.1378 segundos
    - com k = 4 e 10 iterações o tempo de processamento foi de 0.1657 segundos
    - com k = 5 e 10 iterações o tempo de processamento foi de 0.1950 segundos

- **c2ds3-2g.txt**

- com k = 2 e 10 iterações o tempo de processamento foi de 0.1049 segundos

- com  $k = 3$  e 10 iterações o tempo de processamento foi de 0.1282 segundos
- com  $k = 4$  e 10 iterações o tempo de processamento foi de 0.1657 segundos
- com  $k = 5$  e 10 iterações o tempo de processamento foi de 0.2739 segundos

- **monkey.txt**

- com  $k = 5$  e 10 iterações o tempo de processamento foi de 1.2394 segundos
- com  $k = 6$  e 10 iterações o tempo de processamento foi de 0.9086 segundos
- com  $k = 7$  e 10 iterações o tempo de processamento foi de 0.9877 segundos
- com  $k = 8$  e 10 iterações o tempo de processamento foi de 1.1505 segundos
- com  $k = 9$  e 10 iterações o tempo de processamento foi de 1.2065 segundos
- com  $k = 10$  e 10 iterações o tempo de processamento foi de 1.4622 segundos
- com  $k = 11$  e 10 iterações o tempo de processamento foi de 1.4890 segundos
- com  $k = 12$  e 10 iterações o tempo de processamento foi de 1.5795 segundos

- **Single-link**

- **c2ds1-2sp.txt** - com intervalo de 2 a 5 clusters, o tempo de processamento foi de 09.8060 segundos.
- **c2ds3-2g.txt** - com intervalo de 2 a 5 clusters, o tempo de processamento foi de 09.3309 segundos.
- **monkey.txt** - com intervalo de 5 a 12 clusters, o tempo de processamento foi de 09 min e 40.8840 segundos.

- **Average-link**

- **c2ds1-2sp.txt** - com intervalo de 2 a 5 clusters, o tempo de processamento foi de 07.2103 segundos.
- **c2ds3-2g.txt** - com intervalo de 2 a 5 clusters, o tempo de processamento foi de 07.4209 segundos.
- **monkey.txt** - com intervalo de 5 a 12 clusters, o tempo de processamento foi de 07 min e 30.3250 segundos.

## Resultados

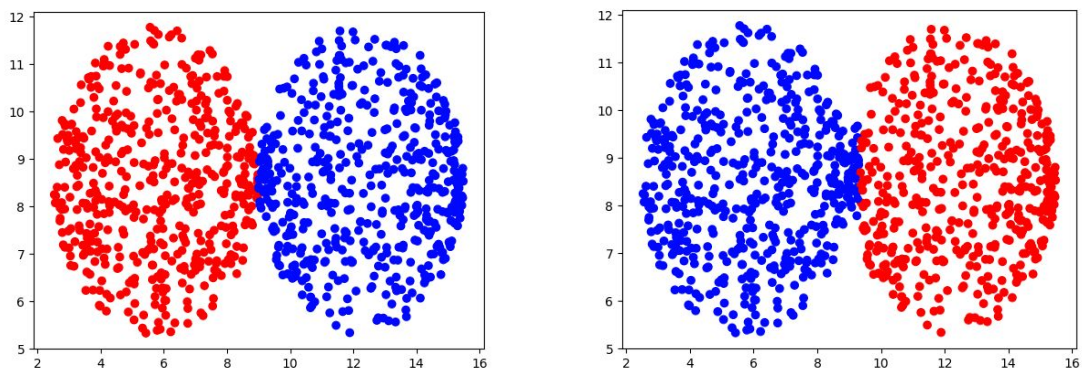
Como esperado, o algoritmo k médias foi o mais rápido, seguido pelo average link e por último o single link. Essa mudança de velocidade acontece pela complexidade dos algoritmos, a linguagem e estruturas de dados usadas e a quantidade de iterações.

O k-médias se comportou melhor no segundo dataset porque esse dataset não tem dados encadeados, mas sim em dados centralizados.

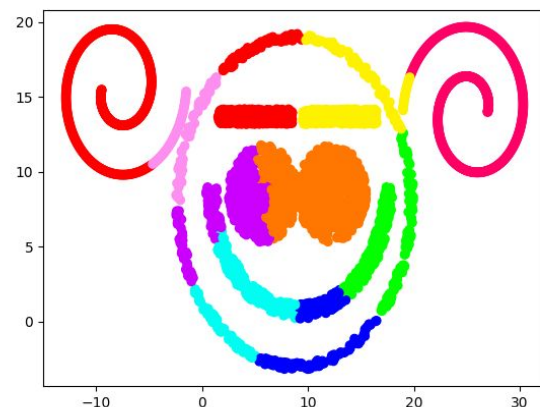
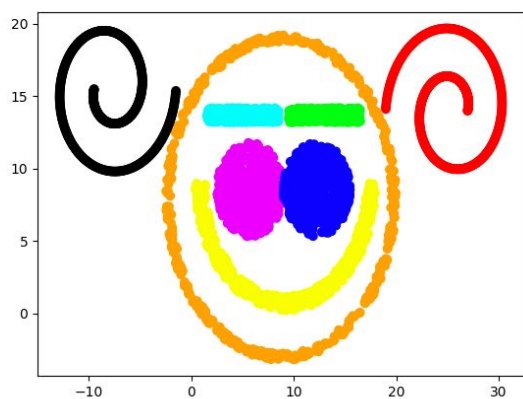
O single link se comportou melhor para o primeiro dataset, uma vez que os dados estão encadeados e portanto a linkagem por menor distância é mais eficiente.

O average-link se comporta bem na maioria dos casos, mas principalmente com dados com características similares.

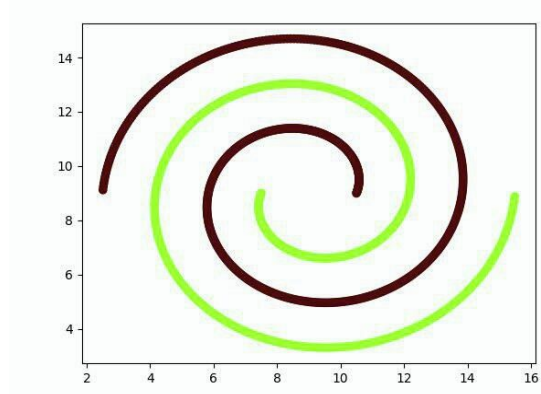
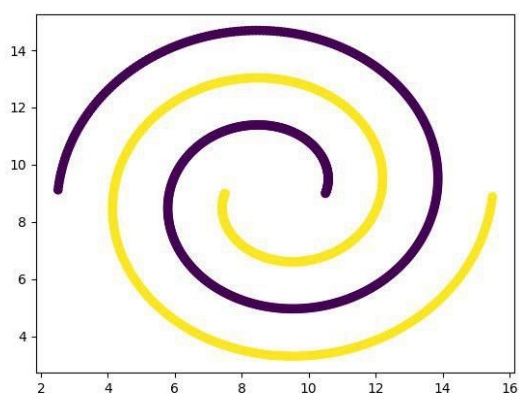
A imagem da esquerda é o algoritmo original enquanto a imagem da direita é o resultado do kmeans implementado, que tem índice rand de 0.8



A figura à esquerda é a figura que representa os dados reais e a figura à direita representa a disposição dos clusters após rodarmos o average-link.



A figura a esquerda representa a partição real e a figura da direita apresenta o resultado do single-link, que foi bastante alto.



A figura à esquerda representa a partição real e a figura da direita apresenta o resultado do single link, que obteve um resultado perfeito, com índice rand 1.