Alunos: Anna Paula Meneghelli de Oliveira
Vinícius Hansen
Disciplina: Teoria dos Grafos – TEG0002
Professor: Gilmário B. Santos

# RELATÓRIO DO DESENVOLVIMENTO DE UM ALGORITMO PARA AGRUPAMENTO DAS ESPÉCIES DA BASE IRIS DATASET UTILIZANDO GRAFOS

## 1. INTRODUÇÃO

O objetivo deste trabalho foi realizar um estudo prático sobre a implementação de um grafo representado por uma estrutura de dados. O estudo prático em si consistiu em separar os tipos de espécies de Iris – *setosa*, *versicolor* e *virgínica* - encontradas no *Iris dataset* [1] utilizando agrupamento de dados.

Grafos são estruturas de dados compostas por vértices e arestas, no qual as arestas interligam dois vértices. Além disso, os grafos também podem conter atributos que descrevem os vértices e as arestas [2]. Uma das formas de criar um grafo é a partir de uma base de dados ou de um conjunto de objetos. Pode-se tomar os itens da base de dados como os vértices do grafo, e definir as arestas por uma relação de semelhança entre os itens (vértices).

Quando o objetivo é separar os itens de uma base de dados em grupos coerentes, pode-se definir a semelhança entre os objetos como uma distância e utilizá-la para realizar um agrupamento de dados. Neste caso, utiliza-se uma função de distância para realizar a divisão do conjunto de dados. Sendo assim, podemos criar um grafo no qual a função de distância define as arestas. Desta forma, o grafo terá agrupamentos nos quais os vértices do mesmo grupo estarão “próximos” uns dos outros e “distantes” de vértices de outros grupos [3].

Neste contexto, o aprendizado de máquinas – subcampo da inteligência artificial que estuda o desenvolvimento de algoritmos capazes de aprender a partir de dados [4] - pode ser usado para tomar decisões sobre a melhor distância a ser utilizada para o processo de agrupamento. Este método também pode ser usado para o agrupamento de casos não conexos, que não foram direcionados a um grupo específico a partir da função de distância inicial. Além disso, é possível avaliar o agrupamento feito a partir de métricas

de avaliação de aprendizado de máquinas, como a matriz de confusão [5], para averiguar a efetividade do algoritmo.

Uma matriz de confusão é uma tabela que pode ser usada para avaliar o desempenho de um algoritmo de classificação. Ela resume visualmente o desemprenho das previsões de um modelo em relação à suas classes reais, de forma que as linhas da tabela representam as classes reais e as colunas representam as classes previstas. As previsões são organizadas em verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo. Um bom modelo resultará em uma alta taxa de verdadeiros positivos e de verdadeiros negativos [6].

## 2. DESENVOLVIMENTO DO ALGORÍTMO E AGRUPAMENTO DOS DADOS

O algoritmo foi desenvolvido em duas linguagens de programação, C e Python. A função de distância e o agrupamento foram desenvolvidos em C, já a matriz de confusão e o histograma foram desenvolvidos em Python.

### 2.1 Agrupamento

A princípio, os casos do banco de dados foram tomados como vértices de um grafo. A distância entre cada vértice, encontrada pelas medidas de tamanho das sépalas e pétalas das flores, foi utilizada para criar matrizes de adjacência para diferentes grafos (trabalho 1).

Para a escolha do limiar de distância mais propício para a o agrupamento, foi feita uma iteração indo de 0,002 a 0,3, somando 0,002 a cada iteração, dando 150 casos de análise. Inicialmente esta iteração havia sido feita de 0,001 a 0,3, iterando a cada 0,001, mas apareceram grupos iguais que foram descartados no tipo de iteração escolhida ao final. Para cada um desses limiares iterados, um grafo é gerado, sendo assim, foi possível analisar 150 grafos diferentes e escolher os mais propícios para o passo seguinte.

A análise do agrupamento foi feita utilizando um algoritmo de busca em profundidade. Este método foi aplicado como uma varredura, ou seja, com o objetivo de percorrer todos os vértices adjacentes a um primeiro vértice inicial. Se os diferentes agrupamentos não possuem arestas entre uns e outros, é possível encontrar todos os vértices pertencentes a um grupo. Também é possível encontrar vértices que não conexos do grafo.

O critério utilizado para a escolha dos grafos foi pegar os que possuíram pelo menos 3 grupos com mais de 20 vértices e menos de 50. Estes parâmetros foram escolhidos pois a base de dados é conhecida. Ela possui 3 espécies de flores, e cada grupo de espécies possuí 50 casos. Sendo assim, um agrupamento com mais de 50 casos contém duas espécies, já um agrupamento com menos de 20 casos é relativamente pequeno comparado à quantidade de casos de cada espécie da flor.

Os limiares que geraram grafos que atendem ao critério foram 0,054, 0,056 e 0,058. Os agrupamentos gerados e os vértices não conexos podem ser vistos na Tabela 1.

| | **Limiar 0,054** | **Limiar 0,056** | **Limiar 0,058** |
|---|---|---|---|
| *Grupo 1* | 0 4 7 9 1 2 3 6 11 24 29 12 25 30 34 35 28 106 5 18 44 46 17 19 21 26 23 39 20 27 31 36 48 16 37 40 49 45 47 8 13 38 42 43 32 33 15 | 0 4 7 9 1 2 3 6 11 17 10 5 18 19 16 33 15 32 46 21 26 23 31 20 27 28 35 12 25 29 24 30 34 39 37 40 49 45 47 8 13 38 42 48 36 43 44 | 0 4 7 9 1 2 3 6 11 17 10 5 16 19 21 26 23 31 20 27 28 35 12 25 29 24 30 34 39 37 40 49 45 47 8 13 38 42 46 32 33 15 44 48 36 43 18 |
| *Grupo 2* | 50 52 76 54 58 65 51 56 85 75 74 97 71 82 67 69 53 80 81 79 89 92 94 55 66 84 78 61 88 95 96 99 63 73 91 90 86 77 | 50 52 76 54 58 65 51 56 85 75 74 97 61 78 55 66 84 96 67 69 53 80 81 79 89 59 82 71 92 94 88 95 99 90 63 73 91 86 77 | 50 52 76 54 58 65 51 56 85 75 74 97 61 71 82 67 69 53 80 81 79 89 59 92 94 55 66 84 96 88 95 99 78 63 73 91 90 86 77 |
| *Grupo 3* | 70 127 123 72 133 83 101 113 121 142 149 138 126 146 111 128 103 116 137 147 110 115 145 112 120 124 140 104 132 143 144 139 141 148 136 | 70 127 123 72 133 83 101 113 121 142 149 138 126 146 111 103 116 128 104 124 112 120 139 141 145 115 110 147 137 136 148 140 143 144 132 | 70 127 123 72 133 83 101 113 121 142 149 138 126 146 111 103 116 128 104 124 112 120 102 125 129 139 141 145 115 110 147 137 136 148 140 143 144 132 |
| *Vértices não conexos* | 14 22 41 57 93 60 59 62 64 68 87 98 100 102 105 122 106 107 130 108 109 114 117 118 119 125 129 131 134 135 | 14 22 41 57 93 60 98 62 64 68 87 100 102 125 129 105 122 106 107 130 108 109 114 117 118 119 131 134 135 | 14 22 41 57 93 60 98 62 64 68 87 100 105 122 106 107 130 108 109 114 117 118 119 131 134 135 |

Tabela 1: Agrupamentos iniciais.

Após isso, foi necessário agrupar os vértices não conexos do grafo a um dos grupos. Para cada vértice não conexo, verificou-se a distância média entre o vértice e

todos os vértices de um dos grupos. Feito isso para os três grupos, o vértice foi inserido no agrupamento cuja distância média entre o grupo e o vértice é a menor. Sempre que um novo vértice é inserido em um grupo, essas distâncias são recalculadas, isso permite que a precisão seja mantida conforme os agrupamentos aumentam de tamanho. O resultado deste processo pode ser visto na Tabela 2.

| | **Limiar 0,054** | **Limiar 0,056** | **Limiar 0,058** |
|---|---|---|---|
| *Grupo 1* | 0 4 7 9 1 2 3 6 11 24 29 12 25 30 34 35 28 10 5 18 44 46 17 19 21 26 23 39 20 27 31 36 48 16 37 40 49 45 47 8 13 38 42 43 32 33 15 14 22 41 | 0 4 7 9 1 2 3 6 11 17 10 5 18 19 16 33 15 32 46 21 26 23 31 20 27 28 35 12 25 29 24 39 37 40 49 45 47 8 13 38 42 48 36 43 44 14 22 41 | 0 4 7 9 1 2 3 6 11 17 10 5 16 19 21 26 23 31 20 27 28 35 12 25 29 24 30 34 39 37 40 49 45 47 8 13 38 42 46 32 33 15 44 48 36 43 18 14 22 41 |
| *Grupo 2* | 50 52 76 54 58 65 51 56 85 75 74 97 71 82 67 69 53 80 81 79 89 92 94 55 66 84 78 61 88 95 96 99 63 73 91 90 86 77 57 93 60 59 62 64 68 87 98 106 119 | 50 52 76 54 58 65 51 56 85 75 74 97 61 78 55 66 84 96 67 69 53 80 81 79 89 59 82 71 92 94 88 95 99 90 63 73 91 86 77 57 93 60 98 62 64 68 87106 119 | 50 52 76 54 58 65 51 56 85 75 74 97 61 71 82 67 69 53 80 81 79 89 59 92 94 55 66 84 96 88 95 99 78 63 73 91 90 86 77 57 93 60 98 62 64 68 87 106 119 |
| *Grupo 3* | 70 127 123 72 133 83 101 113 121 142 149 138 126 146 111 128 103 116 137 147 110 115 145 112 120 124 140 104 132 143 144 139 141 148 136 100 102 105 122 107 130 108 109 114 117 118 125 129 131 134 135 | 70 127 123 72 133 83 101 113 121 142 149 138 126 146 111 103 116 128 104 124 112 120 139 141 145 115 110 147 137 136 148 140 143 102 125 129 105 122 107 130 108 109 114 117 118 131 134 135 | 70 127 123 72 133 83 101 113 121 142 149 138 126 146 111 103 116 128 104 124 112 120 102 125 129 139 141 145 115 110 147 137 136 148 140 143 144 132 100 105 122 107 130 108 109 114 117 118 131 134 135 |

Tabela 2: Agrupamento final.

### 2.2 Análise dos resultados

A análise dos resultados consistiu em verificar se os casos foram agrupados corretamente. Inicialmente, foi verificada a quantidade de espécies do tipo *virgínica*, *setosa* e *versicolor* em cada um dos agrupamentos, como mostra a Figura 1. De acordo com os histogramas apresentados na figura, todos os modelos tiveram uma boa

distribuição em relação ao agrupamento das espécies. Os grupos ficaram com aproximadamente 50 casos cada, o que condiz com a distribuição real da base de dados.

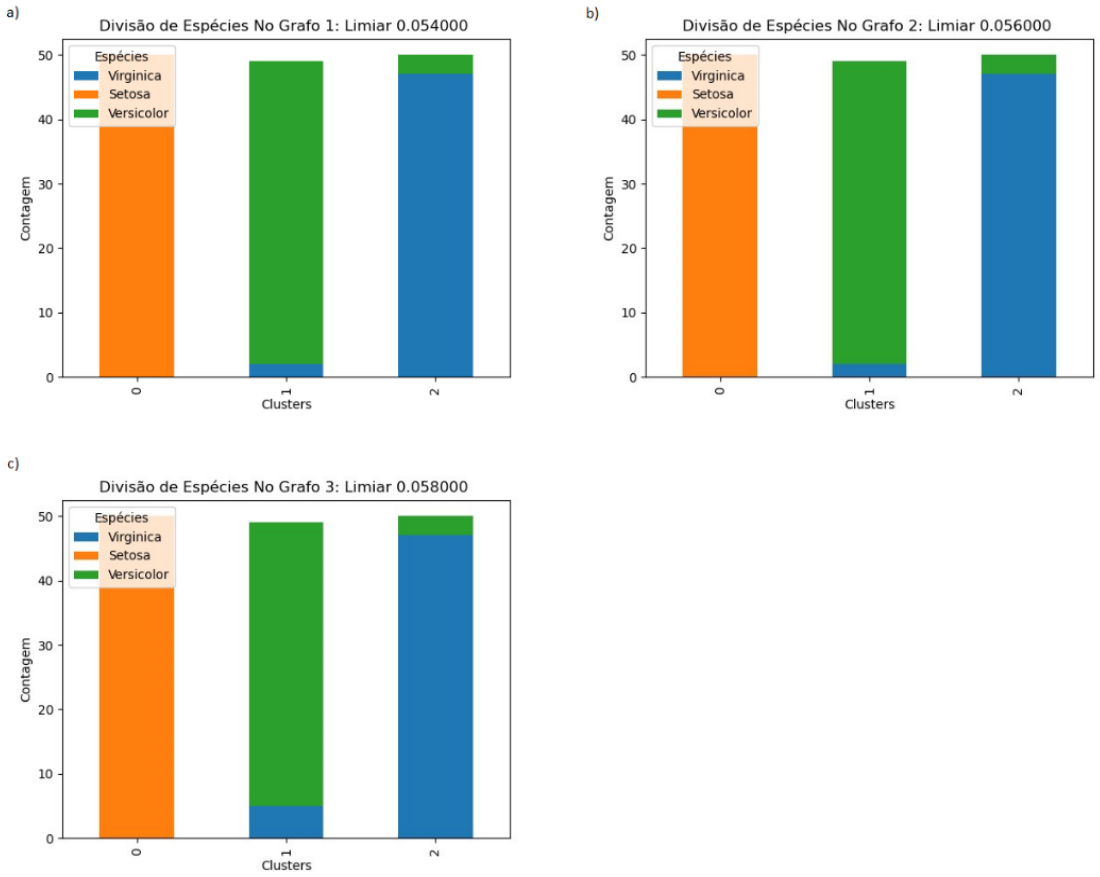


Figura 1: Resultado dos agrupamentos das espécies. a) Para o grafo cujo limiar foi 0,054. b) Para o grafo cujo limiar foi 0,056. c) Para o grafo cujo limiar foi 0,058.

Após esta identificação, foi utilizada uma matriz de confusão para a avaliação do desempenho do algoritmo. Os elementos de cada agrupamento foram classificados em verdadeiro positivo, quando foi previsto que o elemento pertence ao grupo e ele realmente pertence a ele; em verdadeiro negativo, quando foi previsto que o elemento não pertence ao grupo e ele realmente não pertencem a ele; em falso positivo, quando foi previsto que o elemento pertence ao grupo, mas ele não pertence; e em falso negativo, quando foi previsto que o elemento não pertence ao grupo, mas ele pertence. Na Figura 2 estão as matrizes de confusão construídas para os três limiares.

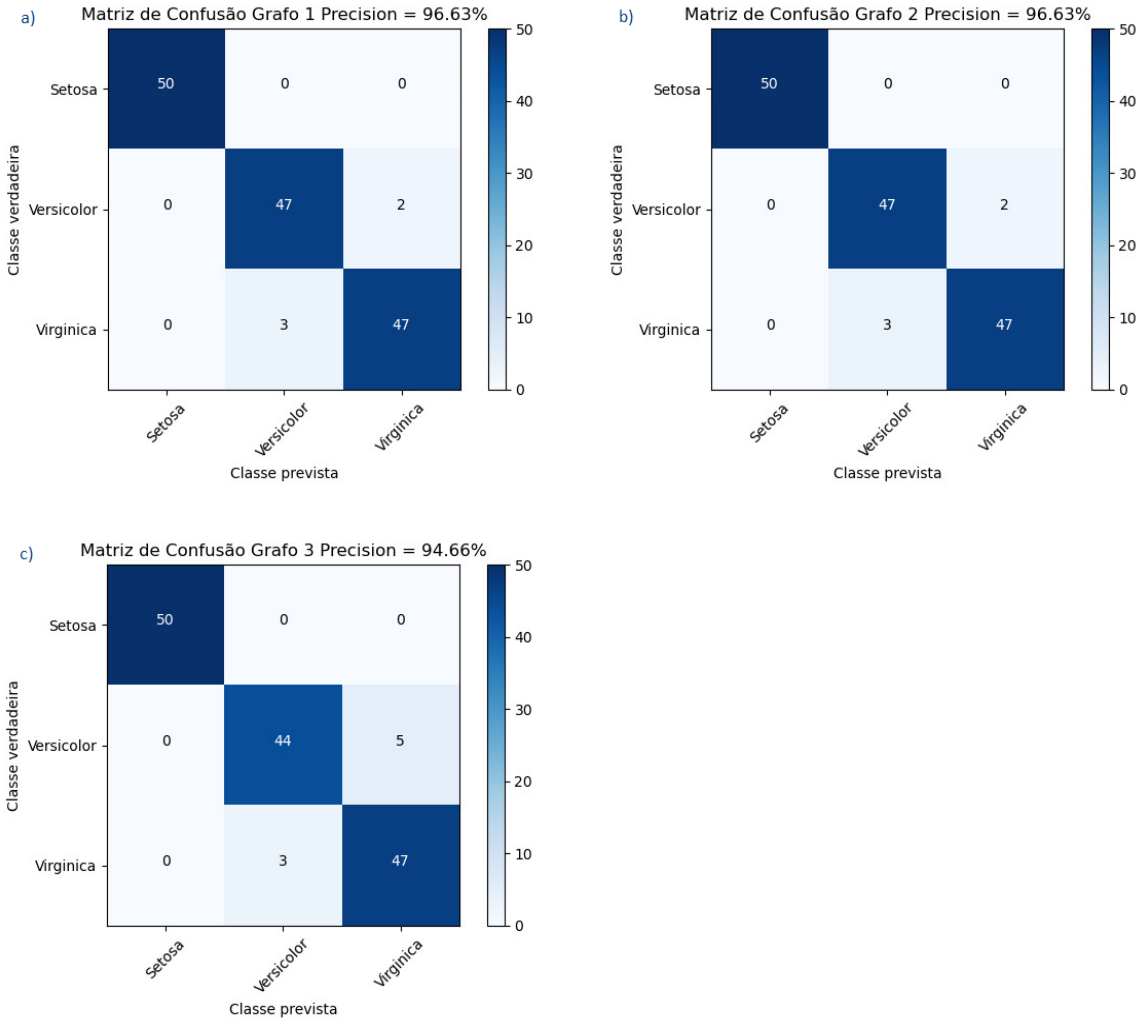


Figura 2: Matriz de confusão. a) Matriz de confusão para o grafo criado com o limiar 0,054. b) Matriz de confusão para o grafo criado com o limiar 0,056. c) Matriz de confusão para o grafo criado com o limiar 0,058.

Percebe-se, pela Figura 2, que o algoritmo utilizado apresentou um bom desempenho. O primeiro e o segundo caso, na Figura 2a e 2b respectivamente, apresentaram precisão de 96,63%, já o terceiro caso, na Figura 2c, apresentou precisão de 94,66%. O resultado foi satisfatório, mas diferente do esperado, pois de acordo com a Tabela 1, o grafo 3 é o que possuí menor quantidade de vértices não conexos, e foi o que obteve a pior precisão.

REFERÊNCIAS

[1] UCI Machine Learning Repository. Iris Data Set [recurso eletrônico]. Irvine: University of California, School of Information and Computer Science, 1988. Disponível em: https://archive.ics.uci.edu/ml/datasets/iris. Acesso em: 05 maio 2023.

[2] XAVIER, Otávio. Uma introdução às Redes Neurais para Grafos (GNN) [recurso eletrônico]. Medium, 2021. Disponível em: https://medium.com/@otaviocx/uma-introdu%C3%A7%C3%A3o-%C3%A0s-redes-neurais-para-grafos-gnn-60e53fcd77d6. Acesso em: 05 maio 2023.

[3] WIKIPÉDIA. Clustering [recurso eletrônico]. São Paulo: Wikimedia Foundation, 2023. Disponível em: https://pt.wikipedia.org/wiki/Clustering. Acesso em: 05 maio 2023.

[4] WIKIPÉDIA. Aprendizado de máquina [recurso eletrônico]. São Paulo: Wikimedia Foundation, 2023. Disponível em: https://pt.wikipedia.org/wiki/Aprendizado_de_m%C3%A1quina. Acesso em: 05 maio 2023.

[5] DATA HACKERS. Entendendo o que é matriz de confusão com Python [recurso eletrônico]. Medium, 2021. Disponível em: https://medium.com/data-hackers/entendendo-o-que-%C3%A9-matriz-de-confus%C3%A3o-com-python-114e683ec509. Acesso em: 05 maio 2023.

[6] ANALYTICS VIDHYA. What is a Confusion Matrix? [recurso eletrônico]. Medium, 2020. Disponível em: https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5. Acesso em: 05 maio 2023.