ÁRVORES DE DECISÃO

FCUP

Trabalho 3

Inteligência Artificial

19-05-2022

Ana Catarina da Silva Costa Gomes (up201804545)
Cláudia da Costa Maia (up201905492)
Maria Santos Sobral (up201906946)

ÍNDICE

Ana Catarina da Silva Costa Gomes (up201804545)
Cláudia da Costa Maia (up201905492)
Maria Santos Sobral (up201906946)

INTRODUÇÃO

Uma árvore de decisão representa uma função que tem como input um vetor constituído por vários atributos e retorna como output uma determinada decisão, que é calculada com base numa sequência de testes. Cada nó interno da árvore corresponde a um teste para um certo valor de um dado atributo e as suas arestas são etiquetadas com os possíveis valores para o atributo em questão.

Deste modo, uma instância é classificada inicialmente pela raiz, que especifica um atributo em particular. De seguida, desce-se na árvore através do ramo que corresponde ao valor do atributo, repetindo-se o processo para a subárvore originada no novo nó. Assim, cada folha na árvore representa uma decisão final, ou seja, um valor a ser retornado pela função.

Além de ser simples e fácil de implementar, este método é muito útil, uma vez que nos permite tomar uma decisão de forma prática, através da comparação das diferentes opções existentes, tendo em conta os seus custos, probabilidades e benefícios.

A árvore de decisão é uma forma de representação do conhecimento, ilustrando os diferentes níveis de importância dos atributos, que podem não estar presentes nas árvores mais compactas, não necessariamente menos assertivas, dada a irrelevância daquela informação para a classificação pretendida.

Como já se adivinha, existem várias árvores possíveis para representar dado problema e a estrutura de cada uma depende, além do conjunto de exemplos de treino inicial, da técnica utilizada pelo agente, sendo de grande interesse elaborar um algoritmo que desenhe uma árvore de decisão que simplifique a leitura e posterior compreensão da classificação sem comprometer a taxa de erro em larga escala

ALGORITMOS E MÉTRICAS UTILIZADAS

Um algoritmo guloso utilizado em árvores de decisão está desenhado para minimizar a profundidade da árvore final. A ideia é escolher sempre o atributo que é mais útil para classificar os exemplos, para que seja necessário utilizar o menor número de atributos possível. Torna-se então necessário arranjar uma forma de medir a utilidade de cada atributo, o que pode ser feito através do ganho de informação, o qual está diretamente relacionado com a entropia. A entropia caracteriza a pureza de um conjunto aleatório de exemplos, isto é, revela a taxa de erros para cada atributo possível num dado nó.

Sejam v_1,…, v_n os valores possíveis para um dado atributo, e P(v_1), …, P(v_n) as suas probabilidades, então a sua entropia é definida como:

$$\mathrm{I}\,(\mathrm{P}(v_1), \ldots, \mathrm{P}(v_n)) \;=\; \sum_{i=1}^{n} -\,\mathrm{P}(v_i)\, log_2(v_i)$$

Existem, no entanto, outras métricas utilizadas para selecionar os atributos a colocar na árvore, tais como o Gini Index, que favorece partições maiores, enquanto a entropia favorecia partições menores. Este índice pode ser calculado através da seguinte fórmula:

Ana Catarina da Silva Costa Gomes (up201804545)
Cláudia da Costa Maia (up201905492)
Maria Santos Sobral (up201906946)

$$1 - \sum_{i=1}^{n} P(v_i)^2$$

Sendo que a impureza do atributo é máxima quando o Gini Index é igual a 1 e mínima quando é igual a 0.

Para além das métricas escolhidas, existem também diferentes algoritmos que podem ser utilizados, sendo que aquele que iremos implementar neste trabalho é o ID3 (Iterative Dichotomiser 3), o qual sugere que se use como métrica a entropia. Em cada iteração deste algoritmo, é calculada a entropia I(S) para todos os atributos do conjunto S que ainda não foram analisados, para que se possa selecionar aquele que tem a menor entropia. No caso de haver um empate, escolhe-se o atributo que tem o menor número de valores distintos, para gerar uma árvore menos densa. O conjunto S é então dividido de acordo com o atributo escolhido, dando origem a vários subconjuntos. O algoritmo continua recursivamente a ser aplicado para cada subconjunto originado, considerando apenas os atributos que ainda não foram utilizados. Esta recursão termina quando todos os elementos de um subgrupo pertencem à mesma classe ou quando já todos os atributos foram analisados. Assim, cada nó interno da árvore gerada representa o atributo selecionado para dividir os dados, enquanto as folhas da árvore representam uma classe, que corresponde ao subconjunto final do ramo em questão.

No entanto, o algoritmo ID3 apresenta algumas limitações, nomeadamente o facto de funcionar apenas para variáveis discretas e para atributos com o mesmo peso. O algoritmo C4.5 é considerado seu sucessor, sendo que funciona de forma muito semelhante, permitindo, no entanto, corrigir alguns dos seus problemas. O C4.5 é capaz de trabalhar com varáveis continuas e também com atributos que tenham diferentes pesos. Além disso, é capaz de funcionar com dados em falta e tem ainda um mecanismo de poda da árvore.

Existe uma versão mais atualizada do algoritmo C4.5, chamada C5.0. Este algoritmo, além de ser significativamente mais rápido e eficiente, elimina os atributos que não são relevantes, conseguindo atingir resultados semelhantes ao C4.5 com uma árvore de decisão muito mais pequena.

Temos também o algoritmo CART (Classification and Regression Trees), que utiliza como métrica o Gini Index, enquanto o C5.0 utilizava critérios baseados no ganho de informação. No algoritmo CART os testes são sempre binários, enquanto no C4.5 o número de ramos da árvore de decisão não é fixo. Além disso, o CART, tal como o C4.5, constrói a árvore de decisão antes da poda, usando um modelo de complexidade de custo. Já o C5.0 realiza a poda após construir a árvore de decisão, através dos limites de confiança binomiais.

DESCRIÇÃO DA IMPLEMENTAÇÃO

A linguagem de programação escolhida foi o Java, uma vez que é a linguagem com que tivemos mais contacto até ao momento e consequentemente, aquela em que nos sentimos mais confortáveis a programar.

Quanto ao código em si, começamos por criar uma classe DT para representar as árvores de decisão, uma vez que à medida que vamos avançando no algoritmo, vão surgindo novas subárvores. Em relação aos nós, já vimos que representam um atributo (no caso de serem nós internos) ou uma decisão final (no caso das folhas), pelo que decidimos criar dois construtores nesta classe, um para os atributos e outro para as folhas.

Optamos então por criar a classe Atributo, que por sua vez tem como atributos o seu nome, um array para representar os seus valores possíveis e um array que regista quantos exemplos existem para cada valor possível. Além disso, ainda nesta classe, decidimos criar um ArrayList chamado *regists* que guarda os registos em si. Optamos por utilizar esta estrutura de dados, uma vez que não sabemos de início quantos registos temos, torna-se muito útil poder ir acrescentando-os à medida que os vamos lendo. (Por este motivo, utilizamos esta estrutura de dados várias vezes ao longo do programa.) Existe ainda uma variável *discretized* em forma booleana, que nos indica se o atributo em questão já foi discretizado. Na classe Atributo criamos ainda um construtor para inicializar cada atributo e também as funções *values* e *register*, para nos auxiliar a guardar os dados iniciais, sendo que esta última tem 2 versões diferentes para tratar o caso de já não ser a primeira vez que se está a registar e, entretanto, ter discretizado os possíveis valores. Já os nós folha, além de conterem o valor da decisão final, têm também um counter, que indica quantos registos da tabela vão ter, durante a descida na árvore, a decisão em questão.

No ficheiro principal (IA03) temos a parte de tratamento de dados que consiste na leitura e armazenamento de cada linha do ficheiro, procedendo-se, em seguida, ao registo adequado de cada atributo presente na tabela. Como podem aparecer nos dados variáveis numéricas com vários valores possíveis (mais do que 6), procede-se à sua discretização tendo por base um algoritmo supervisionado. Conforme foi explicado na aula, este algoritmo consiste na definição de conjuntos de valores possíveis de acordo com a classificação, isto é decisão final. Supondo que a lista dos registos se encontra ordenada pelo atributo em causa, marca-se o início de um conjunto com o valor do atributo do registo que ainda não foi processado e o fim do conjunto com o valor do atributo do k-ésimo registo que tinha uma decisão diferente. Onde k é o limite de "erros" permitidos dentro de um conjunto, constante definida à priori. De seguida, passa-se para a construção da árvore de decisão e, depois de a ter, lê-se e guarda-se os registos de uma tabela de teste, sobre a qual o algoritmo mostrará uma classificação com base na árvore, percorrendo-a, de acordo com os valores dos atributos relevantes até encontrar uma folha. No caso de não ser possível decidir, imprime-se uma mensagem de erro. No fim, é apresentada uma taxa de erro, composta pela quantidade de exemplos de teste sobre os quais não foi possível decidir ou a decisão não era a mesma (isto se os exemplos de teste forem os mesmos que a tabela que deu origem à árvore).

Por sua vez, a construção da árvore foi feita com base no algoritmo ID3, disponível no livro seguido nesta UC, que consiste, de um modo geral, em escolher o atributo mais relevante e dividir a tabela de exemplos consoante os valores possíveis desse atributo e para cada subconjunto constrói-se uma subárvore, pelo mesmo método. Como critério de paragem da execução recursiva do algoritmo tem-se 3 casos base: caso os exemplos tenham todos a mesma decisão, então mais nenhum atributo é relevante para a decisão e a subárvore é constituída por uma única folha com a respetiva classificação; caso não

haja mais atributos ou exemplos a subárvore é constituída por uma única folha indicando a decisão mais frequente quer dos exemplos atuais, se os houver, quer dos exemplos anteriores. Dada a subjetividade da escolha do atributo mais relevante optou-se por escolher o que tinha a menor entropia, métrica aconselhada para este trabalho.

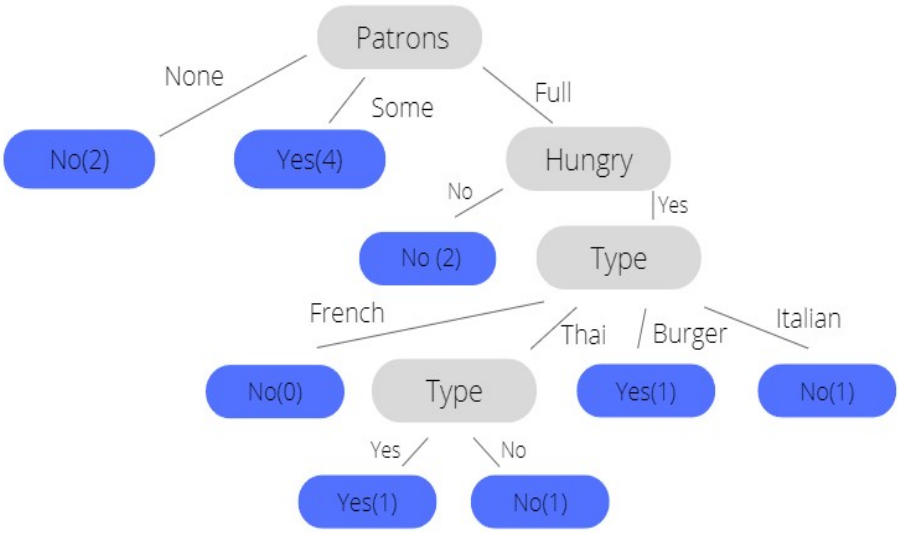
RESULTADOS

Aplicando o algoritmo para cada umas das bases de dados fornecidas obtemos os seguintes resultados:

Dataset – Restaurant:

Árvore de Decisão:

```
<Pat>
  Some: Yes(4)
  Full:
    <Hun>
      Yes:
        <Type>
          French: No(0)
          Thai:
            <Fri>
              No: No(1)
              Yes: Yes(1)
          Burger: Yes(1)
          Italian: No(1)
      No: No(2)
  None: No(2)
```



Fase de testes:

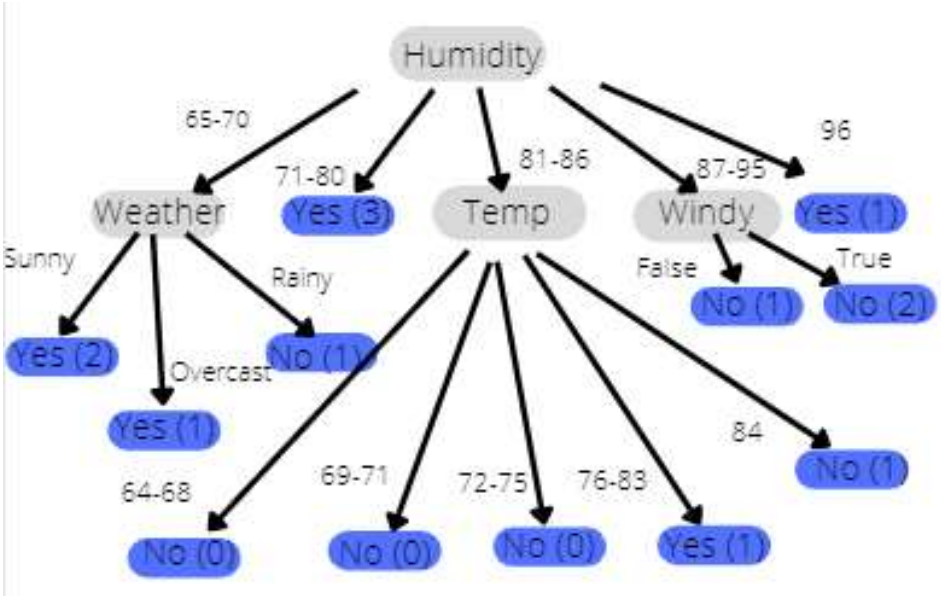
```
[1] Decisão: Yes -- Decisão correta
[2] Decisão: No -- Decisão correta
[3] Decisão: Yes -- Decisão correta
[4] Decisão: Yes -- Decisão correta
[5] Decisão: No -- Decisão correta
[6] Decisão: Yes -- Decisão correta
[7] Decisão: No -- Decisão correta
[8] Decisão: Yes -- Decisão correta
[9] Decisão: No -- Decisão correta
[10] Decisão: No -- Decisão correta
[11] Decisão: No -- Decisão correta
[12] Decisão: Yes -- Decisão correta

Taxa de erro = 0.0
```

Ana Catarina da Silva Costa Gomes (up201804545)
Cláudia da Costa Maia (up201905492)
Maria Santos Sobral (up201906946)

Dataset – Weather:

Árvore de Decisão:

```
<Humidity>
  65 - 70:
    <Weather>
      sunny: yes(2)
      overcast: yes(1)
      rainy: no(1)
  71 - 80: yes(3)
  81 - 86:
    <Temp>
      64 - 68: no(0)
      69 - 71: no(0)
      72 - 75: no(0)
      76 - 83: yes(1)
      84: no(1)
  87 - 95:
    <Windy>
      FALSE: no(1)
      TRUE: no(2)
  96: yes(1)
```



Fase de testes:

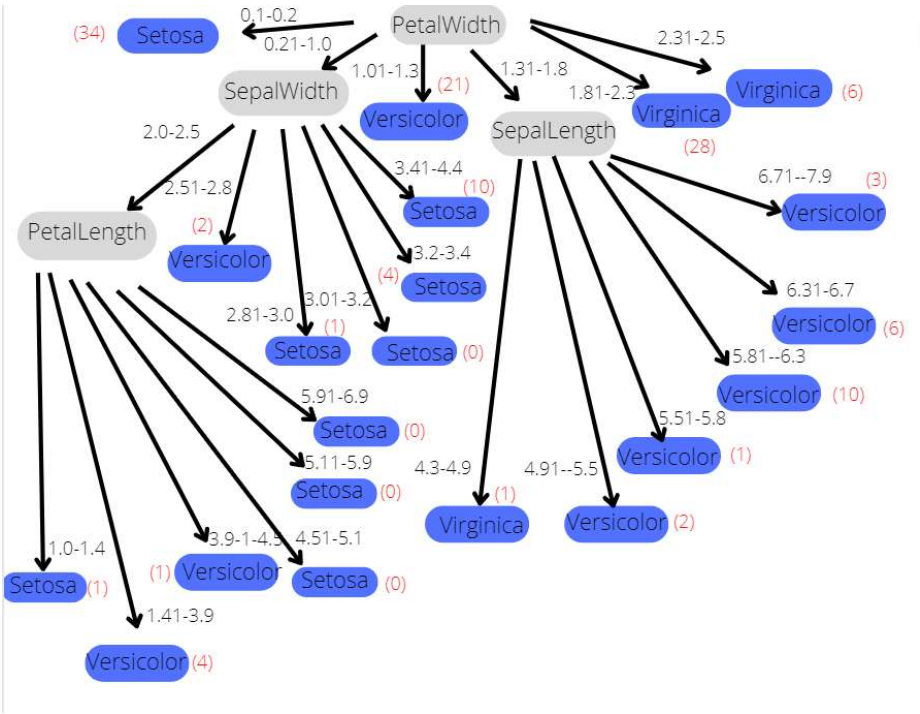
```
[1] Decisão: no -- Decisão correta
[2] Decisão: no -- Decisão correta
[3] Decisão: yes -- Decisão correta
[4] Decisão: yes -- Decisão correta
[5] Decisão: yes -- Decisão correta
[6] Decisão: no -- Decisão correta
[7] Decisão: yes -- Decisão correta
[8] Decisão: no -- Decisão correta
[9] Decisão: yes -- Decisão correta
[10] Decisão: yes -- Decisão correta
[11] Decisão: yes -- Decisão correta
[12] Decisão: no -- Decisão incorreta
[13] Decisão: yes -- Decisão correta
[14] Decisão: no -- Decisão correta

Taxa de erro = 0.07142857142857142
```

Ana Catarina da Silva Costa Gomes (up201804545)
Cláudia da Costa Maia (up201905492)
Maria Santos Sobral (up201906946)

Dataset – Iris:

Árvore de Decisão:

```
<petalwidth>
  0.1 - 0.2: Iris-setosa(27)
  0.21 - 1.0:
    <sepalwidth>
      2.0 - 2.5:
        <petallength>
          1.0 - 1.4: Iris-setosa(1)
          1.41 - 3.9: Iris-versicolor(4)
          3.91 - 4.5: Iris-setosa(0)
          4.51 - 5.1: Iris-setosa(0)
          5.11 - 5.9: Iris-setosa(0)
          5.91 - 6.9: Iris-setosa(0)
      2.51 - 2.8: Iris-versicolor(1)
      2.81 - 3.0: Iris-setosa(2)
      3.01 - 3.2: Iris-setosa(2)
      3.21 - 3.4: Iris-setosa(7)
      3.41 - 4.4: Iris-setosa(11)
  1.01 - 1.3: Iris-versicolor(23)
  1.31 - 1.8:
    <sepallength>
      4.3 - 4.9: Iris-virginica(1)
      4.91 - 5.5: Iris-versicolor(2)
      5.51 - 5.8: Iris-versicolor(1)
      5.81 - 6.3: Iris-versicolor(10)
      6.31 - 6.7: Iris-versicolor(6)
      6.71 - 7.9: Iris-versicolor(3)
  1.81 - 2.3: Iris-virginica(30)
  2.31 - 2.5: Iris-virginica(13)
```



Fase de testes:

```
[1] Decisão: Iris-setosa -- Decisão correta
[2] Decisão: Iris-setosa -- Decisão correta
[3] Decisão: Iris-setosa -- Decisão correta
[4] Decisão: Iris-setosa -- Decisão correta
[5] Decisão: Iris-setosa -- Decisão correta
[6] Decisão: Iris-setosa -- Decisão correta
[7] Decisão: Iris-setosa -- Decisão correta
[8] Decisão: Iris-setosa -- Decisão correta
[9] Decisão: Iris-setosa -- Decisão correta
[10] Decisão: Iris-setosa -- Decisão correta
[11] Decisão: Iris-setosa -- Decisão correta
[12] Decisão: Iris-setosa -- Decisão correta
[13] Decisão: Iris-setosa -- Decisão correta
[14] Decisão: Iris-setosa -- Decisão correta
[15] Decisão: Iris-setosa -- Decisão correta
[16] Decisão: Iris-setosa -- Decisão correta
[17] Decisão: Iris-setosa -- Decisão correta
[18] Decisão: Iris-setosa -- Decisão correta
[19] Decisão: Iris-setosa -- Decisão correta
[20] Decisão: Iris-setosa -- Decisão correta
```

```
[21] Decisão: Iris-setosa -- Decisão correta
[22] Decisão: Iris-setosa -- Decisão correta
[23] Decisão: Iris-setosa -- Decisão correta
[24] Decisão: Iris-setosa -- Decisão correta
[25] Decisão: Iris-setosa -- Decisão correta
[26] Decisão: Iris-setosa -- Decisão correta
[27] Decisão: Iris-setosa -- Decisão correta
[28] Decisão: Iris-setosa -- Decisão correta
[29] Decisão: Iris-setosa -- Decisão correta
[30] Decisão: Iris-setosa -- Decisão correta
[31] Decisão: Iris-setosa -- Decisão correta
[32] Decisão: Iris-setosa -- Decisão correta
[33] Decisão: Iris-setosa -- Decisão correta
[34] Decisão: Iris-setosa -- Decisão correta
[35] Decisão: Iris-setosa -- Decisão correta
[36] Decisão: Iris-setosa -- Decisão correta
[37] Decisão: Iris-setosa -- Decisão correta
[38] Decisão: Iris-setosa -- Decisão correta
[39] Decisão: Iris-setosa -- Decisão correta
[40] Decisão: Iris-setosa -- Decisão correta
[41] Decisão: Iris-setosa -- Decisão correta
[42] Decisão: Iris-setosa -- Decisão correta
[43] Decisão: Iris-setosa -- Decisão correta
[44] Decisão: Iris-setosa -- Decisão correta
```

```
[45] Decisão: Iris-setosa -- Decisão correta
[46] Decisão: Iris-setosa -- Decisão correta
[47] Decisão: Iris-setosa -- Decisão correta
[48] Decisão: Iris-setosa -- Decisão correta
[49] Decisão: Iris-setosa -- Decisão correta
[50] Decisão: Iris-setosa -- Decisão correta
[51] Decisão: Iris-versicolor -- Decisão correta
[52] Decisão: Iris-versicolor -- Decisão correta
[53] Decisão: Iris-versicolor -- Decisão correta
[54] Decisão: Iris-versicolor -- Decisão correta
[55] Decisão: Iris-versicolor -- Decisão correta
[56] Decisão: Iris-versicolor -- Decisão correta
[57] Decisão: Iris-versicolor -- Decisão correta
[58] Decisão: Iris-versicolor -- Decisão correta
[59] Decisão: Iris-versicolor -- Decisão correta
[60] Decisão: Iris-versicolor -- Decisão correta
[61] Decisão: Iris-versicolor -- Decisão correta
[62] Decisão: Iris-versicolor -- Decisão correta
[63] Decisão: Iris-versicolor -- Decisão correta
[64] Decisão: Iris-versicolor -- Decisão correta
[65] Decisão: Iris-versicolor -- Decisão correta
[66] Decisão: Iris-versicolor -- Decisão correta
[67] Decisão: Iris-versicolor -- Decisão correta
[68] Decisão: Iris-versicolor -- Decisão correta
```

Ana Catarina da Silva Costa Gomes (up201804545)
Cláudia da Costa Maia (up201905492)
Maria Santos Sobral (up201906946)

```
[69] Decisão: Iris-versicolor -- Decisão correta
[70] Decisão: Iris-versicolor -- Decisão correta
[71] Decisão: Iris-versicolor -- Decisão correta
[72] Decisão: Iris-versicolor -- Decisão correta
[73] Decisão: Iris-versicolor -- Decisão correta
[74] Decisão: Iris-versicolor -- Decisão correta
[75] Decisão: Iris-versicolor -- Decisão correta
[76] Decisão: Iris-versicolor -- Decisão correta
[77] Decisão: Iris-versicolor -- Decisão correta
[78] Decisão: Iris-versicolor -- Decisão correta
[79] Decisão: Iris-versicolor -- Decisão correta
[80] Decisão: Iris-versicolor -- Decisão correta
[81] Decisão: Iris-versicolor -- Decisão correta
[82] Decisão: Iris-versicolor -- Decisão correta
[83] Decisão: Iris-versicolor -- Decisão correta
[84] Decisão: Iris-versicolor -- Decisão correta
[85] Decisão: Iris-versicolor -- Decisão correta
[86] Decisão: Iris-versicolor -- Decisão correta
[87] Decisão: Iris-versicolor -- Decisão correta
[88] Decisão: Iris-versicolor -- Decisão correta
[89] Decisão: Iris-versicolor -- Decisão correta
[90] Decisão: Iris-versicolor -- Decisão correta
[91] Decisão: Iris-versicolor -- Decisão correta
[92] Decisão: Iris-versicolor -- Decisão correta
```

```
[93] Decisão: Iris-versicolor -- Decisão correta
[94] Decisão: Iris-versicolor -- Decisão correta
[95] Decisão: Iris-versicolor -- Decisão correta
[96] Decisão: Iris-versicolor -- Decisão correta
[97] Decisão: Iris-versicolor -- Decisão correta
[98] Decisão: Iris-versicolor -- Decisão correta
[99] Decisão: Iris-versicolor -- Decisão correta
[100] Decisão: Iris-versicolor -- Decisão correta
[101] Decisão: Iris-virginica -- Decisão correta
[102] Decisão: Iris-virginica -- Decisão correta
[103] Decisão: Iris-virginica -- Decisão correta
[104] Decisão: Iris-versicolor -- Decisão incorreta
[105] Decisão: Iris-virginica -- Decisão correta
[106] Decisão: Iris-virginica -- Decisão correta
[107] Decisão: Iris-virginica -- Decisão correta
[108] Decisão: Iris-versicolor -- Decisão incorreta
[109] Decisão: Iris-versicolor -- Decisão incorreta
[110] Decisão: Iris-virginica -- Decisão correta
[111] Decisão: Iris-virginica -- Decisão correta
[112] Decisão: Iris-virginica -- Decisão correta
[113] Decisão: Iris-virginica -- Decisão correta
[114] Decisão: Iris-virginica -- Decisão correta
[115] Decisão: Iris-virginica -- Decisão correta
[116] Decisão: Iris-virginica -- Decisão correta
```

```
[117] Decisão: Iris-versicolor -- Decisão incorreta
[118] Decisão: Iris-virginica -- Decisão correta
[119] Decisão: Iris-virginica -- Decisão correta
[120] Decisão: Iris-versicolor -- Decisão incorreta
[121] Decisão: Iris-virginica -- Decisão correta
[122] Decisão: Iris-virginica -- Decisão correta
[123] Decisão: Iris-virginica -- Decisão correta
[124] Decisão: Iris-versicolor -- Decisão incorreta
[125] Decisão: Iris-virginica -- Decisão correta
[126] Decisão: Iris-versicolor -- Decisão incorreta
[127] Decisão: Iris-versicolor -- Decisão incorreta
[128] Decisão: Iris-versicolor -- Decisão incorreta
[129] Decisão: Iris-virginica -- Decisão correta
[130] Decisão: Iris-versicolor -- Decisão incorreta
[131] Decisão: Iris-virginica -- Decisão correta
[132] Decisão: Iris-virginica -- Decisão correta
[133] Decisão: Iris-virginica -- Decisão correta
[134] Decisão: Iris-versicolor -- Decisão incorreta
[135] Decisão: Iris-versicolor -- Decisão incorreta
[136] Decisão: Iris-virginica -- Decisão correta
[137] Decisão: Iris-virginica -- Decisão correta
[138] Decisão: Iris-versicolor -- Decisão incorreta
[139] Decisão: Iris-versicolor -- Decisão incorreta
[140] Decisão: Iris-virginica -- Decisão correta
```

```
[141] Decisão: Iris-virginica -- Decisão correta
[142] Decisão: Iris-virginica -- Decisão correta
[143] Decisão: Iris-virginica -- Decisão correta
[144] Decisão: Iris-virginica -- Decisão correta
[145] Decisão: Iris-virginica -- Decisão correta
[146] Decisão: Iris-virginica -- Decisão correta
[147] Decisão: Iris-virginica -- Decisão correta
[148] Decisão: Iris-virginica -- Decisão correta
[149] Decisão: Iris-virginica -- Decisão correta
[150] Decisão: Iris-versicolor -- Decisão incorreta

Taxa de erro = 0.1
```

Ana Catarina da Silva Costa Gomes (up201804545)
Cláudia da Costa Maia (up201905492)
Maria Santos Sobral (up201906946)

CONCLUSÕES

Com base nos resultados obtidos, podemos verificar que quanto mais acima na árvore estão os atributos, mais importantes são na decisão, e por sua vez também a entropia é menor (o ganho da informação é maior). Assim, verificamos que a compacidade da árvore resulta de vários fatores, sendo o primeiro a entropia que verificamos de facto que é uma boa métrica. Por sua vez, se houver empate no cálculo da entropia, optamos por escolher o atributo com menos valores possíveis de forma a reduzir o número de ramos da árvore. Por último, também com o objetivo de ter menos ramos, procedemos à discretização das variáveis numéricas de modo a terem no máximo 6 valores possíveis em cada atributo.

Em adição, para avaliar a qualidade da árvore, no sentido de fazer um balanço entre ser compacidade e assertividade, calculamos a taxa de erro no fim da fase de testes, de modo a podermos avaliar esta última característica. Notamos que no primeiro caso a taxa de erro era de 0%, o que é suportado pela evidência de que a soma dos counters, apresentados em cada folha terminal da árvore, é precisamente igual ao número de amostras. Isto é, no processo de descida da árvore todos os registos do ficheiro de testes vão terminar numa decisão e essa quantidade de registos que termina numa dada folha de decisão é precisamente o que representa o counter. Sobre este dataset, podemos ainda notar a influência da métrica para a escolha do atributo mais relevante, no caso a entropia, que nos permitiu passar de uma representação dos dados numa tabela com 10 atributos para uma árvore com 4 e com assertividade máxima para os 12 exemplos de teste, mostrando que havia muitos atributos irrelevantes para a classificação pretendida.

Já os dois casos seguintes apresentam uma taxa de erro não nula, sendo no caso do dataset Weather aproximadamente igual a 7% e para o dataset Iris aproximadamente igual a 10%. Um dos possíveis motivos para a taxa de erro apresentada para Weather é o tamanho reduzido da amostra, pois, apesar de apenas falhar 1 exemplo de teste, num total de 14 exemplos já assume uma importância considerável. Tendo em conta que também possui 2 atributos numéricos sobre os quais se realiza a discretização, o tamanho do conjunto de treino aliado com a forte variação do valor desses atributos numéricos nesses mesmos exemplos contribui para a incapacidade de obter uma árvore compacta e assertiva. De salientar que o único caso em que decide incorretamente se deve á falta de atributos para dividir o conjunto, uma vez que continua a existir exemplos de classes distintas no mesmo nó, mas já não temos nenhum atributo que nos permita dividir o conjunto. Algo que é suportado pelo facto de a soma dos counters não dar o total dos exemplos de treino. Portanto, é visível a influência da quantidade e respetiva qualidade dos dados no resultado obtido.

No último caso, o dataset Iris, a taxa de erro verificada permitiu-nos concluir que se devia ao facto dos atributos numéricos, nos quais se registou bastantes valores possíveis, terem sido discretizados. A condição de ter no máximo 6 valores possíveis por atributo, obrigando à discretização em caso contrário, pode estar a afetar a assertividade da árvore, verificando-se o tal caso de ter exemplos de classes distintas num mesmo nó que inevitavelmente se torna terminal, por já não ter mais atributos para dividir o conjunto, optando pela decisão mais frequente. Podemos, então, notar a influência da técnica de discretização usada, que possivelmente seria melhor, para a taxa de erro, se a quantidade máxima de classes diferentes dentro de um mesmo conjunto fosse mais pequena.

Ana Catarina da Silva Costa Gomes (up201804545)
Cláudia da Costa Maia (up201905492)
Maria Santos Sobral (up201906946)

Para além disso, verificamos uma possível melhoria para a compacidade da árvore que, no entanto, aumentaria a taxa de erro. Esta melhoria consiste em menosprezar a existência de classificações diferentes no mesmo conjunto, que para além de estar em minoria, a sua percentagem é residual.

Alem disto, no caso de testarmos a árvore com um conjunto de testes diferentes, como existia a possibilidade de não conseguirmos concluir nenhuma decisão, torna-se necessário refazer a árvore acrescentando o exemplo que falhou aos dados iniciais. A partir daqui existem pelo menos duas formas de corrigir a árvore, fazendo a aprender a nova amostra de forma indutiva ou incremental, isto é, pedindo uma nova árvore de raiz (indutiva) ou apenas acrescentar uma nova subárvore sem precisar de induzir todas as amostras já analisadas.

Em conclusão, tentamos equilibrar a compacidade da árvore com a sua assertividade, sendo que esta última se pode verificar com a taxa de erros. Os resultados que obtivemos dependem, como vimos, da métrica para a escolha do atributo mais relevante, do tipo, quantidade e qualidade dos dados de que dispomos para treinar a árvore de decisão e ainda da técnica de discretização das variáveis numéricas, bem como do limite máximo de classes que cada atributo deste tipo pode ter.

Ana Catarina da Silva Costa Gomes (up201804545)
Cláudia da Costa Maia (up201905492)
Maria Santos Sobral (up201906946)

REFERÊNCIAS BIBLIOGRÁFICAS

• Slides da cadeira Inteligência Artificial (CC2006) do ano letivo 2021/2022

• Artificial Intelligence: a Modern Approach, by Stuart Russell and Peter Norvig, 3rd edition, Prentice Hall

• Data Mining: Practical Machine Learning Tools and Techniques, Ian H. Witten, Eibe Frank, Mark A. Hall, 3rd edition

Ana Catarina da Silva Costa Gomes (up201804545)
Cláudia da Costa Maia (up201905492)
Maria Santos Sobral (up201906946)