**Universidade do Minho**
Escola de Ciências

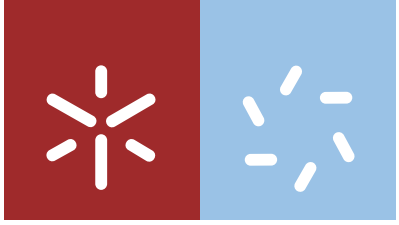
Paulina Da Silva Orlando Suquina

# Estudo e Construção de árvores de decisão: aplicação ao ensino

outubro de 2019

Paulina Da Silva Orlando Suquina **Estudo e Construção de árvores de decisão: aplicação ao ensino**

UMinho | 2019



**Universidade do Minho**
Escola de Ciências

Paulina Da Silva Orlando Suquina

# Estudo e Construção de árvores de decisão: aplicação ao ensino

Dissertação de Mestrado
Mestrado em Matemática e Computação

Trabalho efetuado sob a orientação da
**Professor Doutor Stéphane Louis Clain**

outubro de 2019

## DIREITOS DO AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

# Agradecimetos

Primeiramente agradeço a Deus por ter me dado força e coragem para enfrentar as dificuldades.
Não posso deixar de agradecer ao meu orientador, Professor Doutor Stéphane Louis Clain, por toda paciência, empenho, motivação com que sempre me orientou neste trabalho. Obrigada por me corrigir sempre que foi necessário.
Agradeço também aos Professores do departamento do Mestrado em Matemática e Computação, que mesmo com a chegada tardia a Universidade, deram-nos o apoio incondicional de que precisávamos. A Doutora Paula Henriques, pela oportunidade que me deu de poder fazer parte deste projeto.
Desejo igualmente agradecer aos meus colegas, em especial o Incansável Pedro Vicente, Schields Pedro, a irmã que ganhei em Braga Maria Tómas, Gerson Hungulu, Osvaldo Jamba e Nunes Rafael, cujo apoio e amizade estiveram presentes em todos os momentos.
Agradeço ao meu esposo Eduardo Nangacovie, pelo incentivo, apoio e por acreditar que eu era capaz de concluir este trabalho. Ao meu filho David Nangacovie, por suportar a minha ausência por varias horas ao longo do dia e por vezes aos fins de semana. Agradeço também a Tia Maria Silva, Por cuidar do David nos momentos que mais precisei.
A Tia Helena Canhici, por ter me indicado as pessoas certas cá em Braga.
Aos meus irmãos pelo amor, incentivo e apoio incondicional.
Ao meu Pai Domingos Suquina, pelo amor, dedicação e por ter apostado na minha formação.
Agradeço a minha mãe Irene Henriques da Silva, meu porto seguro que sempre me apoiou nas horas mais difíceis de desânimo e cansaço.
Por ultimo agradeço a minha família e amigos, pelo apoio incondicional que me deram.

## DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

# Resumo

As árvores de decisão são ferramentas muito utilizadas em áreas como as de Extração de Conhecimento de Dados (ECD), devido à eficiência que elas possuem em produzir classificadores. As mesmas são vantajosas devido à sua capacidade em dividir um espaço de exemplos em subespaços, e ajustar cada subespaço recorrendo a diferentes modelos de classificação. Este trabalho pretende fazer um estudo relativamente à construção de árvores de decisão utilizando diferentes técnicas de pré-poda, que têm como finalidade melhorar a qualidade de um classificador. Assim, com a utilização de uma Base de Dados (BD) real ligada à área do ensino, referente a escola secundaria Conde de Monsaraz, à qual pertence ao agrupamento vertical de Escolas de Reguengos de Monsaraz, são feitas várias experiências com diferentes critérios de paragem, obtendo como resultado duas Matrizes de Confusão (MC) referentes aos dados de treino e de teste.

Assim, a utilização de indicadores como o *Recall* e a *Especificity*, que são adequados ao problema em causa, possibilitam a quantificação do erro do classificador. No final das experiências obtém-se um gráfico que corresponde ao valor do indicador vs o critério de paragem utilizado. Desta forma, o resultado deste gráfico são duas curvas, uma associada aos dados de treino e outra associada aos dados de teste.

**Palavras-chave**: Árvore de decisão, Poda, Pré-poda, Classificação, Matriz de Confusão.

# Abstract

Decision trees are widely used as inference tools in areas such as Data Extraction, due to their efficiency in producing classifiers. Their ability to partition the attribute space into subspaces labeled with class values. This work aims at studing the construction of decision trees using different pruning techniques to improve the quality and the efficiency of a classifier. We shall apply the methodology to real Databases connected to the teaching area, namely the secondary school Conde de Monsaraz. Several experiments were carried out with different stopping criteria, to provide two Confusion Matrices (for the training and test dataset) that enable the accuracy of the method.

More specifically, indicators such as *Recall* and *Especificity* are appropriate to our real problem for quantifying the classifier error. At the end of the experiments, a figure displays the correspondance of the indicator vs. the stopping criterion threshold and provide two curves that give a prediction of the most effective decision tree.

**keywords**:Decision tree, Pruning, Pre-pruning, Classification, Confusion matrices.

# Índice Geral

# Lista de Figuras

# Lista de Tabelas

# Siglas

**BD** Base de Dados. iv, 76, 77

**ECD** Extração do Conhecimento de Dados. iv

**FN** *False Negative*. 55

**FP** *False Positive*. 55

**ISE** *In Sample Error*. ix, 78

**MC** Matriz de Confusão. iv

**OSE** *Out Sample Error*. ix, 78

**TN** *True Negative*. 55

**TP** *True Positive*. 55

# Capítulo 1

# Introdução

Atualmente com o aumento do volume de dados gerados em diferentes ramos de atividade, surge a necessidade de serem criadas ferramentas computacionais mais complexas e independentes, capazes de a partir de experiências passadas, criar hipóteses que possam resolver um determinado problema. A tais experiências dão-se o nome de ECD. Assim, as Organizações, têm a vantagem de, a partir da ECD, poderem tomar melhores decisões, com vista a rever decisões anteriores, elaborar estratégias para tomada de decisões futuras, obter melhor conhecimento sobre os dados da organização, entre outras.

A área de ECD apresenta vários métodos para extrair informações, designadamente a classificação,associação, *clustering*, padrões sequenciais, regressão e detecção de desvios.

Neste trabalho, abordaremos sobre o método de classificação. Este método permite obter padrões através da construção de classificadores e, neste caso em particular, as árvores de decisão como uma técnica de representação do conhecimento contido no conjunto de dados analisado.

As árvores de decisão são ferramentas poderosas de classificação consideradas como uma das técnicas mais eficientes aplicadas em vários campos científicos, tais como *Machine Learning* e Inteligência Artificial [EMS02][Qui86]. Elas permitem, com base num conjunto de atributos extremamente diversificados, classificar populações, eventos, produtos, e posteriormente auxiliar na tomada de decisões [Loh11]. Uma árvore de decisão utiliza, uma estratégia

de *dividir-para-conquistar*, onde um problema complexo é decomposto em sub-problemas mais simples. Por sua vez, esta estratégia é aplicada recursivamente a cada sub-problema. Áreas como:

- A Saúde: na aprendizagem de diagnósticos médicos,no controlo de gastos hospitalares, entre outros;
- As Finanças: na aprendizagem e avaliação do risco de crédito dos solicitantes de empréstimo entre outros;
- A Astronomia: em observações atmosféricas, em análise de informações espaciais;
- A Agricultura: na identificação de doenças em produções agrícolas;
- A Gestão de recursos

e muitas outras, utilizam as árvores de decisão, devido à sua flexibilidade, robustez, facilidade de compreensão e velocidade de processamento.

## 1.1 Objetivos

1. Fazer um estudo sobre as várias técnicas de construção de árvores de decisão nas diferentes literaturas apresentadas, para perceber o contexto e expor vantagens em relação a técnicas de Poda adequadas, tomando em conta o tipo de dados específicos da área do ensino.
2. Desenvolver uma *script* para construção de árvore de decisão usando linguagem de programação *Python* e a biblioteca *scikit-learn* para análise dos dados.

## 1.2 Estrutura da Dissertação

Para além do presente capítulo, esta dissertação está organizada em mais 7 capítulos, nomeadamente:

**Capítulo 2**, que fornece alguns conceitos básicos muito utilizados acerca dos dados, tais como a informação, os atributos e as classes. Apresenta-se, igualmente, as noções de níveis de partição associados a atributos, árvore e grafos.

No **Capítulo 3**, são apresentadas as funções de impureza, bem como a noção de frequência e a classificação do melhor atributo por meio do ganho de informação.

No **Capítulo 4**, por meio de dois exemplos, são construídas árvores de decisão, utilizando o ganho como argumento para a construção das mesmas. É também apresentado neste capítulo, a poda elementar utilizando critérios simples, como nós puros ou nós sem elementos.

**Capítulo 5**, em que se aborda sobre a quantificação do erro da árvore de decisão utilizando a MC e indicadores. Para o efeito são usados dados de treinamento e de teste.

**Capítulo 6**, em que se apresenta o uso da poda de uma árvore de decisão, em particular o método de pré-poda e, os seus respetivos critérios de paragem.

As aplicações da técnica de árvores de decisão são detalhadas no **Capítulo 7**, onde é apresentada a aplicação do modelo de classificação, utilizando uma BD da área do ensino.

Por fim, no **capítulo 8** apresentam-se as conclusões finais do trabalho.

# Capítulo 2

# Dados e árvores

No presente trabalho consideram-se dados como sendo equivalente a informação. Os dados podem aparecer em formas diversificadas e esta diversidade torna-os mais ricos e complexos. Um conjunto de dados ou *"dataset"* é um conjunto de dados normalmente organizado em tabelas. Por cada elemento indicam-se várias características. Cada coluna representa uma variável particular. Cada linha corresponde a um determinado membro do conjunto de dados em questão. Cada valor é conhecido como um dado. O conjunto de dados pode incluir dados para um ou mais membros, correspondente ao número de linhas. Ou seja, Um conjunto D de dados é caracterizado por $d(t) = (t, x(t), c(t))$, onde:

- $t$ é o índice, $t = 1, ..., N$

- $d(t)$ é uma ocorrência constituída por: $x(t) = (x_1(t), x_2(t), ..., x_I(t))$ são as variáveis ou atributos e também são considerados como dados de entrada(*input*)

  $c(t)$ é a classe, que é considerada como dado de saída(*output*)

## 2.1 Atributos e Classes

Chamaremos A a um atributo, quando se tratar de dados de entrada(*input*) e C a uma classe quando se tratar de dados de saída(*output*).


Figura 2.1: Atributo-Classe

Sejam $A_1, A_2, ..., A_I$, um conjunto de atributos, notamos por $\mathcal{A} = A_1 \times A_2, ..., \times A_I$, e C uma classe.

Para cada valor de $x_i$ , temos um conjunto $Ai$ de J valores: $A_i = \{a_{i1}, a_{i2}, ...a_{ij}, ..., a_{iJ}\}$.

Para a classe $C$, temos um conjunto de $K$ decisões: $C = c_1, ..., c_k, ..., c_K$

Exemplo1: Dada a tabela 2.1 onde, a última coluna representa a classe:

Tabela 2.1: Tab.Eventos

| Nº | Sexo | TrabalhoM | TrabalhoP | RazãoEsEscola | **ApoioEF** |
|---|---|---|---|---|---|
| 1 | F | Em casa | Professor | Curso | **Não** |
| 2 | F | Em casa | Outro | Curso | **Sim** |
| 3 | F | Em casa | Outro | Outro | **Não** |
| 4 | F | Saúde | Serviços | Casa | **Sim** |
| 5 | F | Outro | Outro | Casa | **Sim** |
| 6 | M | Serviços | Outro | Reputação | **Sim** |
| 7 | M | Outro | Outro | Casa | **Não** |
| 8 | F | Outro | Professor | Casa | **Sim** |
| 9 | M | Serviços | Outro | Casa | **Sim** |
| 10 | M | Outro | Outro | Casa | **Sim** |
| 11 | F | Professora | Saúde | Reputação | **Sim** |
| 12 | F | Saúde | Outro | Reputação | **Sim** |
| 13 | M | Serviços | serviços | Curso | **Sim** |
| 14 | M | Saúde | Outro | Curso | **Sim** |
| 15 | M | Professora | Outro | Casa | **Sim** |

os atributos são:

$A_1$= {F, M}, $J = 2$;

$A_2$= {Em casa, Saúde, Outro, Serviços, Professora}, $J = 5$;

$A_3$= {Saúde, Outro, Serviços, Professor}, $J = 4$;

$A_4$= {Curso, Outro, Casa, Reputação, }, $J = 4$;

$C$= {Sim, Não}, $K = 2$.

Exemplo2: Neste exemplo temos a tabela 2.2, na qual os valores são numéricos:

Tabela 2.2: Tab.Eventos só com atributos

| N° | TEstudo | TEscolaCasa | TLivres | RFamiliar | NAcademiPai |
|---|---|---|---|---|---|
| 1 | 2 | 2 | 3 | 4 | 4 |
| 2 | 2 | 1 | 3 | 5 | 1 |
| 3 | 2 | 1 | 3 | 4 | 1 |
| 4 | 3 | 1 | 2 | 3 | 2 |
| 5 | 2 | 1 | 3 | 4 | 3 |
| 6 | 2 | 1 | 4 | 5 | 3 |
| 7 | 2 | 1 | 4 | 4 | 2 |
| 8 | 2 | 2 | 1 | 4 | 4 |
| 9 | 2 | 1 | 2 | 4 | 2 |
| 10 | 2 | 1 | 5 | 5 | 2 |
| 11 | 2 | 1 | 3 | 3 | 4 |
| 12 | 3 | 3 | 2 | 5 | 4 |
| 13 | 1 | 1 | 3 | 4 | 1 |
| 14 | 2 | 2 | 4 | 5 | 4 |
| 15 | 1 | 1 | 5 | 4 | 3 |

e os atributos são:

$A_1$= {1, 2, 3}, $J = 3$ (Referente a TEstudo);

$A_2$= {1, 2, 3}, $J = 3$ (Referente a TEscolaCasa);

$A_3$= {1, 2, 3, 4, 5}, $J = 5$ (Referente a TLivres);

$A_4$= {1, 2, 3, 4, 5}, $J = 5$ (Referente a RFamiliar);

$A_5$= {1, 2, 3, 4}, $J = 4$ (Referente a NAcademiPai).

Os atributos ou variáveis podem ser de natureza muito diversificada e podem ser classificados segundo[Mar07][McC98] em atributos qualitativos e quantitativos.

**Atributos qualitativos** são aqueles cuja escala de medida apenas indica a sua presença em categorias de classificação discreta exaustivas e mutuamente exclusivas: Eles podem ser Nominais, binárias e Ordinais.

- Nominais: é apenas uma lista não ordenada de valores que não tem qualquer ligação entre eles. Por exemplo, o atributo $Sexo = \{masculino, feminino\}$
- Ordinais: são atributos constituídos por uma lista ordenada, segundo uma relação descritível mas não quantificável. Por exemplo, habilitações literárias= *{Básico, Secundário, Universitário}*

**Atributos quantitativos**: são atributos cuja escala de medida permite a ordenação e quantificação de diferenças entre elas. Podem ser:

- **intervalar**: integram dados quantitativos, numa escala numérica com intervalos iguais, como por exemplo a temperatura medida em graus Celsius ou em graus Faherenheit. Estas escalas não possuem zero absoluto, isto é não possuem uma medida de ausência de atributo.
- **Razão**: assumem uma escala numérica de valores quantitativos cuja relação exata entre estes é possível definir porque esta escala possui um zero absoluto , como por exemplo o peso ou altura.

## 2.2 Partição associada aos atributos

Seja $\mathcal{D}$ um conjunto de dados. Uma partição $\mathcal{P}$ de $\mathcal{D}$ é constituída de subconjuntos $\{D_1, D_2, ..., D_N\}$ tal que:

1. $D_i \bigcap D_j = \varnothing, i \neq j$;
2. $\bigcup_{i=1}^{N} D_i = \mathcal{D}$

Seja $A_i$ um atributo de valores, $\{a_{i1}, a_{i2}, ..., a_{ij}\}$:

$$D_j = (x(t), y(t)).$$

Onde $y(t)$ são os valores das classes

tal que, $x_i(t) = a_{ij}$, $i = 1, ..., I.\ \ j = 1, ..., N$

Por exemplo, a base de dados representada pela tabela 2.1, será particionada pelo atributo; $A_1 = \{F, M\}$ $a_{11} = F, a_{12} = M$ e $\mathcal{P} = \{D_1, D_2\}$ $D_1 = \{1, 2, 3, 4, 5, 8, 11, 12\}$ $D_2 = \{6, 7, 9, 10, 13, 14, 15\}$

A partir dos subconjuntos $D_1, D_2$, podem ser feitas outras partições. Para isso são particionadas cada uma das partições com um outro atributo da base de dados, tendo assim um segundo nível de partição.

Neste caso, a partição $D_1$ do exemplo anterior, será particionada pelo atributo:

$A_4 = \{$Curso, Casa, Reputação, outro$\}$ com os valores $a_{41} =$ Curso, $a_{42} =$ Casa, $a_{43} =$ reputação, $a_{44} =$ Outro.

Assim sendo, teremos as seguintes partições:

$D_{11} = \{1, 2\}$ , $D_{12} = \{4, 5, 8\}$, $D_{13} = \{11, 12\}$, $D_{14} = \{3\}$

A partição $D_2$ será particionada, com o atributo:

$A_2 = \{$Em casa, Saúde, Outro, Serviços, Professora$\}$

com $a_{21} =$ Em casa, $a_{22} =$ Saúde, $a_{23} =$ outro, $a_{24} =$ Serviços, $a_{25} =$ Professora

Assim sendo, teremos as seguintes partições:

$D_{21} = \varnothing$, $D_{22} = \{14\}$, $D_{23} = \{7, 10\}$, $D_{24} = \{6, 9, 13\}$, $D_{25} = \{15\}$

## 2.3 Árvores

Antes de começarmos a falar de árvore, vamos entender o que é um grafo. Um Grafo é uma estrutura de dados não linear, constituído por nós ou vértices e arestas ou ramos,

$$G = (V, A),$$

em que V representa o vértice e A representa a aresta. No qual os vértices ou nós representam dados tais como: Pessoas, Cidades, números etc. e as arestas representam a existência de ligações entre nós.

Os grafos geralmente podem ser representados de várias formas, uma delas é a que vemos na figura abaixo:

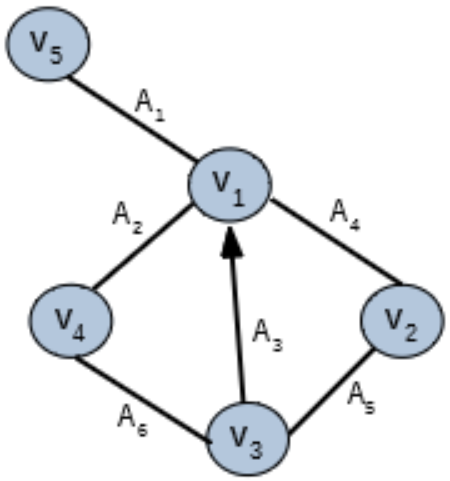



Figura 2.2: Um grafo com 5 vértices e 6 arestas

Na figura 2.2, considerou-se um grafo com cinco vértices e seis arestas, na qual 5 vértices são não orientados: $A_1 = \{v_1v_5\} = \{v_5v_1\}; A_2 = \{v_1v_4\} = \{v_4v_1\}; A_4 = \{v_1v_2\} = \{v_2v_1\}; A_5 = \{v_2v_3\} = \{v_3v_2\}$; $A_6 = \{v_3v_4\} = \{v_4v_3\}$, e um vértice que é orientado: $A_3 = \{v_3v_1\} \neq \{v_1v3\}$.

Entre as diferentes estruturas que podem ser representadas por um grafo, está a Árvore, entendida como um grafo sem ciclo que têm uma origem, o nó raiz. Os nós situados a uma distância n chamam-se nós de nível n e os nós que não possuem ramos são chamados de nó folha ou nó terminal.

Há diversas formas de representação de uma árvore: hierárquica, diagrama de inclusão, diagrama de barras, numeração por níveis, entre outras. Neste trabalho usaremos a forma hierárquica, representada como na figura 2.3:

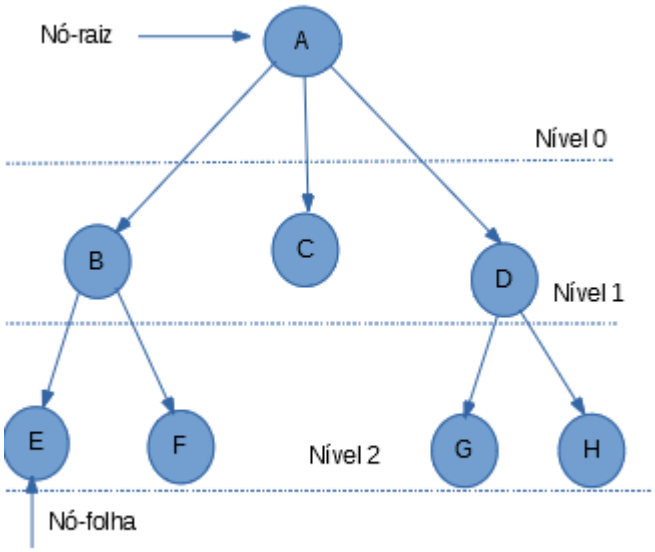



Figura 2.3: árvore

O nível do nó raiz é sempre 0. A altura de um nó é o comprimento do caminho mais longo entre ele e uma folha. Já a profundidade de um nó é a distância percorrida da raiz a este nó. No caso da árvore acima a altura é de 2 e a profundidade é 0 para o nó A e 1 para o nó B.

## 2.4 Representação de partição com uma árvore

Nesta secção, fazemos uma associação entre os nós de uma árvore com uma partição, para melhor visualização destas partições. O nó raiz representa o conjunto total dos dados, o primeiro nível da árvore representa a primeira partição, o segundo nível representa a segunda partição e assim sucessivamente. Está partição é construída á base de um atributo.

A seguir temos o exemplo da construção de uma árvore baseada no exemplo anterior sobre partição do primeiro nível, em que usamos o atributo $A_1$, da base de dados representada pela tabela 2.1:

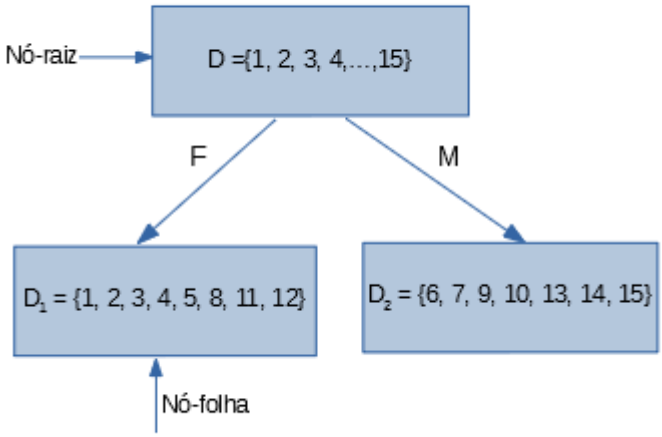



Figura 2.4: Primeiro nível de partição

Depois do primeiro nível de partição, podemos fazer um segundo nível, em que cada ramo terá um outro subconjunto, de um outro atributo. Como exemplo, vamos construir uma segunda partição na árvore, com o atributo $A_4$ para a partição $D_1$ e o atributo $A_3$ para a partição $D2$, referentes a base de dados da tabela 2.1:

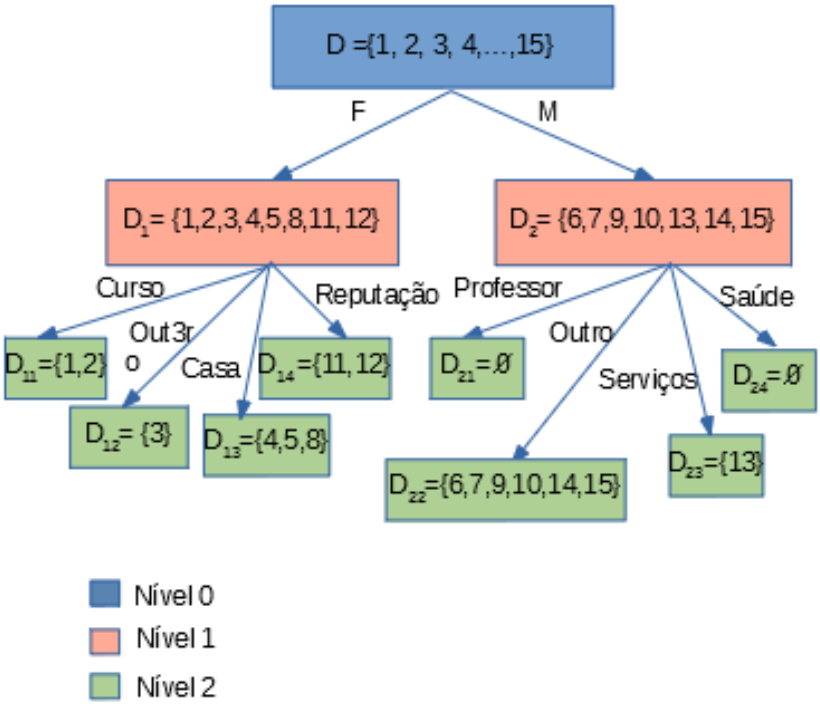



Figura 2.5: Segundo nível de partição

Cada nível é uma partição do conjunto inicial, mas cada uma mais reduzida em relação ás partições anteriores. Por conseguinte, com o mesmo conjunto inicial, podemos fazer diversas árvores. Para a tabela 2.1, com quatro atributos existentes, podemos fazer na ordem de $4!$ árvores.

Até ao momento, as árvores foram associadas apenas aos atributos. Construiremos, em seguida uma árvore de decisão e, para isso, precisaremos de novos elementos para auxiliar na tomada de decisão, tais como a frequência, probabilidade, ganho, etc, que serão abordados no próximo capítulo.

# Capítulo 3

# Probabilidade e Teoria da Informação

Como referenciado, no capítulo anterior utilizamos os atributos para definir partições. Agora, vamos utilizar a informação de classe para definir as probabilidades. As classes podem ser classificadas como binárias, quando apresentam duas categorias, por exemplo {1 e 0}; {sim e não}, como multi-nominais, quando apresentam 3 ou mais categorias, por exemplo {1, 2, 3, 4}; {verde, amarela, Branca, azul}.

## 3.1 Frequências

A probabilidade procura quantificar informações sobre a frequência e a ocorrência ou não de um determinado evento, de uma maneira universal: Seja $D = (w^n)_n$ onde, n = 1,...,N, com $w = (x^n, y^n)$ o conjunto de dados, e seja $E \subset D$ então:

$$P(E) = \frac{|E|}{|D|}$$

$P(E)$, representa a probabilidade de E

$|E|$ representa o conjunto de elementos de E

$|D|$ é o conjunto dos elementos de D.

Seja $C = \{c_1, c_2, ..., c_K\}$ onde $k = 1, ..., K$ e $E(c_k) = \{(x^n, y^n) \in E\}$ tal que, $y^n = c_k$, logo, definimos a frequência associada a $c_k$ por:

$$P_k = P(E(c_k)) = \frac{|E(c_k)|}{|E|}$$

Como $\bigcup E(c_k) = E$ e $E(c_k) \cap E(c_l) = \varnothing$ onde, $l \neq k$ , temos uma partição. Logo: $P = (p_1, p_2, ..., p_k)$, define uma lei de probabilidade.

Nota: Quando não há ambiguidade no ramo $p_k = P(E(k))$, onde $P = (p_1, ..., p_K)$.

A probabilidade varia entre $0 \leq p_k \leq 1$, e de um modo geral,

$$\sum_{k=1}^{K} p_k = 1$$

Com base, mais uma vez, na tabela 2.1, na qual consideramos o atributo 5 como a classe, teremos a distribuição dos dados com as seguintes probabilidades:

Tabela 3.1: Probabilidade com classe binária

| $\|D\| = 15$ | |
|---|---|
| $\|D(\{y = Sim\})\| = 12$ | $\|D(\{y = \text{não}\})\| = 3$ |

Logo, $p_S(D) = \frac{4}{5}$ e $p_N(D) = \frac{1}{5}$ onde, $p_S$ e $p_N$ representam as frequências de Sim e não respetivamente.

Podemos também considerar a tabela 2.2, em que o atributo 2 será a classe, com 3 valores:

Tabela 3.2: Probabilidade com classe multi-nominal numérica

| $\|D\| = 15$ | | |
|---|---|---|
| $\|D(\{y = 1\})\| = 11$ | $\|D(\{y = 2\})\| = 3$ | $\|D(\{y = 3\})\| = 1$ |

logo, $p_1(D) = \frac{11}{15}$; $p_2(D) = \frac{1}{5}$ e $p_3(D) = \frac{1}{15}$ onde, $p_1, p_2$ e $p_3$ são as frequências associadas aos valores 1, 2 e 3 respetivamente.

Em um terceiro exemplo, onde a classe é multi-nominal, relacionada com a qualidade das

relações familiares dos alunos, classificadas em: Bom, Péssimo(PM), Razoável(Raz) e Muito bom(MB):

Tabela 3.3: Probabilidade com classe multi-nominal

| D = 15 | | | |
|---|---|---|---|
| $\|D(\{y = Bom\})\| = 11$ | $\|D(\{y = \text{PM}\})\| = 1$ | $\|D(\{y = \text{Raz}\})\| = 1$ | $\|D(\{y = MB\})\| = 2$ |

Assim, $p_{Bom}(D) = \frac{11}{15}$; $p_{PM}(D) = \frac{1}{15}$; $p_{Raz}(D) = \frac{1}{15}$; $p_{MB}(D) = \frac{2}{15}$. $p_{Bom}$, $p_{PM}$, $p_{Raz}$ e $p_{MB}$, são as frequências associadas aos valores Bom, Péssimo, Razoável e Muito bom.

## 3.2 Noção de Impureza

Nesta secção, avaliaremos a qualidade da informação de um conjunto de dados, a partir do nível de impureza da informação que este apresenta.
Para exemplificar, apresentamos um conjunto $E$ que é constituído pelos elementos F e M, onde, F são os elementos femininos e M os elementos masculinos. Como definida na secção anterior, as probabilidades referentes a estes conjuntos serão:

$$p_F = \frac{|F|}{|E|}, p_M = \frac{|M|}{|E|}$$

Assim sendo, podemos definir uma função de impureza $I$, como uma função que irá satisfazer as seguintes propriedades: $I(E)$ quantifica o nível de impureza de $E$;
$I(E) = 0$ se a $p_F = 1$ e $p_M = 0$ ou o inverso. A função $I(E)$ atingirá o seu máximo quando $p_F = p_M = \frac{1}{2}$.

Deste modo, teremos duas visões de dados diferentes: a visão Microscópica, onde cada elemento caracteriza o conjunto, e a Macroscópica, onde vamos dar uma classificação global a este conjunto. A transformação de uma informação microscópica para uma informação macroscópica acarreta perdas consideráveis de informação, o que nos leva a querermos saber qual é a quantidade de informação que estamos a perder. É neste contexto onde entra a

impureza, que vai avaliar ou quantificar a quantidade de informação que estamos a perder quando passamos a esta transformação.

Seguidamente, é mostrado um exemplo de como quantificar estas perdas, utilizando F como o conjunto de informação macroscópica.

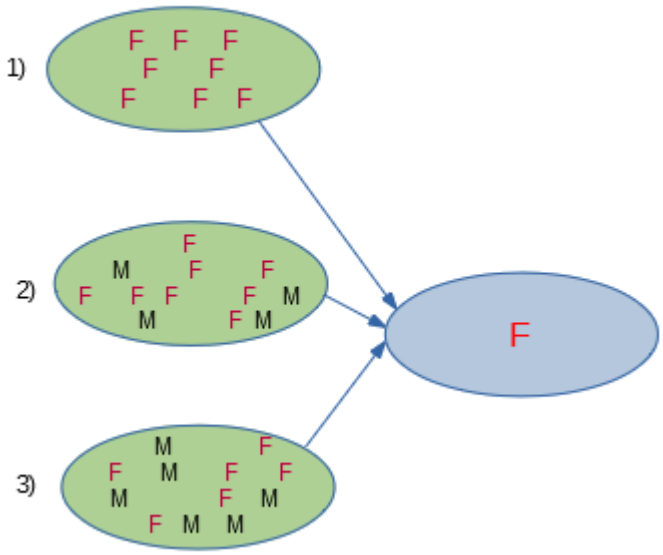



Figura 3.1: Níveis de impureza

Na figura 3.1, temos três situações diferentes de quantificação de impureza:

1. $p_F = 1$ e $p_M = 0$ logo, não há perda de informação porque a classificação é pura, todos os elementos do conjunto pertencem a uma única classe ;
2. $p_F = \frac{2}{3}$ e $p_M = \frac{1}{3}$, logo, o nível de impureza é médio e há uma perda média de $\frac{1}{3}$ da informação;
3. $p_F = p_M = \frac{1}{2}$, atingiu o nível de impureza máximo, visto que estamos a perder metade da informação. É considerada a pior situação.

### 3.2.1 Construção da função impureza

Nesta secção, quantificaremos a noção de impureza na base das frequências.
Dado $E$, um subconjunto da base de dados, a partir deste subconjunto podemos construir $E_1, E_2, ..., E_K$, onde são associadas probabilidades $p_1, p_2, ..., p_K$. Desta matéria, vamos construir uma nova informação, a impureza e que, será representada por três funções:

1. A **entropia** é dada pela fórmula:

$$I_E(E) = -\sum_{k=1}^{K}(p_k log_2 p_k)$$

onde, $E$ é o conjunto de amostras, $p_k$ é a frequência de cada valor da classe, e K é o número de classes;

2. **Índice de Gini** é dado pela fórmula:

$$I_G(E) = 1 - \sum_{k=1}^{K} p_k^2$$

onde $K$ e $p_k$ representam o que foi referenciado no item anterior;

3. **Misclassification**, é dada pela fórmula:

$$I_M(E) = 1 - \max(p_1, ..., p_K)$$

.

Em seguida, representamos graficamente estas três medidas de impureza considerando que $K = 2$, assim, $p_1 + p_2 = 1$ então, $p_2 = 1 - p_1$. Considerando $p_i$, onde, $i = 1, 2, ..$ teremos:

Para a **Entropia** teremos: $I_E(E) = -p_1 ln_2 p_1 - p_2 lg_2 p_2$, teremos: $I_E(E) = -p_1 lg_2 p_1$.

Para o **Índice Gini** teremos:$I_G(E) = 1 - p_1^2 - (1 - p_1)^2 = 2p_1(1 - p_1)$.

Para o **Misclassification** teremos: $I_M(E) = 1 - \max(p_1, p_2) = (p_1, 1 - p_1)$

Sendo assim, a baixo é apresentado o gráfico no qual estão presentes as três funções:

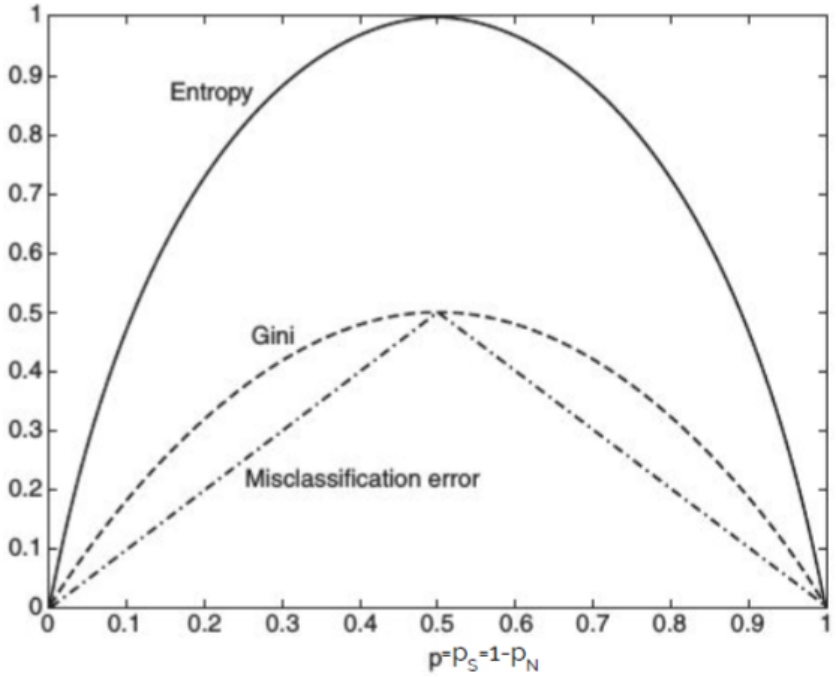



Figura 3.2: Funções de Impureza(adaptado de [GCF+15])

Em seguida, apresentamos exemplos, em que utilizaremos as três funções de impureza. Em um primeiro exemplo, utilizaremos o quinto atributo da tabela 2.1, que para este caso é a classe com dois valores Sim(S) e Não(N). Utilizando as frequências obtidas a partir da tabela 3.1, em que, $p_S = \frac{4}{5}$ e $p_N = \frac{1}{5}$. Assim, temos as seguintes funções:

**Entropia**:$-(4/5)\log_2(4/5) - (1/5)\log_2(1/5) = 0.721$;

**Índice Gini**:$1 - (4/5)^2 - (1/5)^2 = 0.4$;

**Misclassification**:$1 - \max(4/5, 1/5) = 0.2$

Para um segundo exemplo, utilizamos a classe nível de segurança da escola, tirada da base de dados sobre o desempenho dos alunos. Esta possui 365 eventos e a classe tem cinco valores que são: Bom, Mau, Muito bom(MB), péssimo(PS) e Razoável(RZ).

Calculando as frequências: $p_{Bom} = 271/365 = 0.742; p_{Mau} = 13/365 = 0.036;$
$p_{MB} = 39/365 = 0.107; p_{PS} = 14/365 = 0.038; p_{RZ} = 28/365 = 0.077$
Desta forma, calculando as funções teremos:

**Entropia**: $-(0.742)\log_2(0.742) - (0.036)\log_2(0.036) - (0.107)\log_2(0.107) -$
$(0.038)\log_2(0.038) - (0.077)\log_2(0.077) = 1.2995$;

**Índice Gini**: $1 - (0.742)^2 - (0.036)^2 - (0.107)^2 - (0.038)^2 - (0.077)^2 = 0.4287$;

**Misclassification**: $1 - \max(0.742, 0.036, 0.107, 0.038, 0.077) = 0.2575.$

## 3.3 Noção de Ganho da informação

Como vimos anteriormente, utilizamos os atributos para definir partições e as classes para definir probabilidades ou frequências. Nesta secção, vamos escolher um atributo, através do qual faremos uma partição e desta partição calcular qual é o ganho de informação relacionado a este atributo ou partição.

Assim sendo, comecemos por definir o ganho associado a uma partição:
Seja um conjunto $D$, que representa o atributo tempo, particionado em dois conjuntos: o conjunto Sol e o conjunto Chuva. O ganho de informação relacionado ao melhor agrupamento

de Sim(S) e Não(N) deste atributo é representado da seguinte forma:

$$G(D) = I(D) - I_{final}(Tempo)$$

A impureza final é calculada pela fórmula:

$$I_{final}(Tempo) = \frac{|Sol|}{|D|} * I(Sol) + \frac{|Chuva|}{|D|} * I(Chuva)$$

e a impureza inicial é calculada a partir da fórmula da entropia vista anteriormente:

$$I(D) = -\sum_{k=1}^{2}(p_k \log_2 p_k)$$

onde, $k \in \{S, N\}$.

Tendo os conjuntos $D = \{17S, 13N\}, Sol = \{11S, 6N\}, Chuva = \{6S, 7N\}$, podemos calcular o ganho do atributo Tempo como se segue:

Primeiramente calculamos as frequências de cada um deles:
$p_S(D) = \frac{17}{30};\ p_N(D) = \frac{13}{30};\ p_S(Sol) = \frac{11}{17};\ p_N(Sol) = \frac{6}{17};\ p_S(Chuva) = \frac{6}{13};$
$p_N(Chuva) = \frac{7}{13}\ e\ q(Sol) = \frac{17}{30};\ q(Chuva) = \frac{13}{30}$
onde, $q(Sol)\ e\ q(Chuva)$ são as frequências de Sol e chuva em 30 dias. Assim:

$$I(D) = -\frac{17}{30}\log\frac{17}{30} - \frac{13}{30}\log\frac{13}{30} = 0,987$$

.

$$I(Sol) = -\frac{11}{17}\log\frac{11}{17} - \frac{6}{17}\log\frac{6}{17} = 0,936$$

.

$$I(Chuva) = -\frac{6}{13}\log\frac{6}{13} - \frac{7}{13}\log\frac{7}{13} = 0,996$$

Logo:

$$I(Tempo) = 0,936 * \frac{17}{30} + 0,996 * \frac{13}{30} = 0,962$$

$$G(Tempo) = 0,987 - 0,962 = 0,025$$

No exemplo anterior, classificamos o atributo tempo mediante uma classe binária (Sim e Não). Neste exemplo, classificaremos os alunos, que estão representados pelo atributo Sexo, com valores Feminino(F) e Masculino(M). Esta classificação será com base nas notas do primeiro período, utilizando assim uma classe multi-nominal que possui quatro valores: Péssimo(PS), Mau, Bom e Muito Bom(MB).

Assim, os valores serão classificados com base nas ocorrências escritas na figura 3.3:

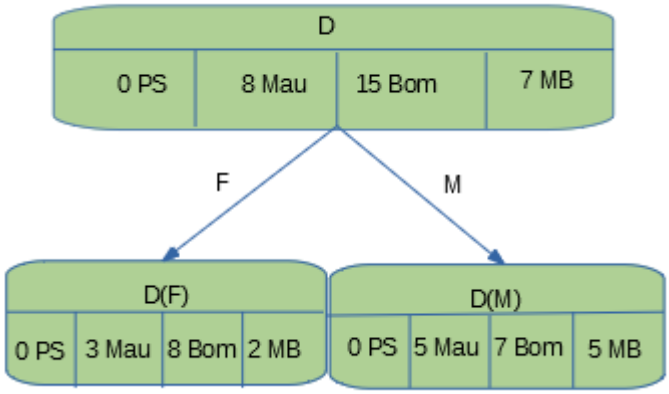



Figura 3.3: Figura das ocorrências utizando uma classe com 4 valores

Como visto anteriormente, para calcularmos o ganho de informação de um atributo, precisamos saber quais são as frequências relacionas a cada uma das ocorrências. Assim, $p_{PS}(D) = 0;\ p_{Mau}(D) = \frac{4}{15};\ p_{Bom}(D) = \frac{1}{2};\ p_{MB}(D) = \frac{7}{30};\ q(F) = \frac{13}{30};\ q(M) = \frac{17}{30};\ p_{PS}(F) = p_{PS}(M) = 0;\ p_{Mau}(F) = \frac{3}{13};\ p_{Bom}(F) = \frac{8}{13};\ p_{MB}(F) = \frac{2}{13};\ p_{Mau}(M) = \frac{5}{17};\ p_{Bom}(M) = \frac{7}{17};\ p_{MB}(M) = \frac{5}{17}$. Logo:

$$I(D) = -\frac{4}{15}\log\frac{4}{15} - \frac{1}{2}\log\frac{1}{2} - \frac{7}{30}\log\frac{7}{30} = 1,498$$

$$I(F) = -\frac{3}{13}\log\frac{3}{13} - \frac{8}{13}\log\frac{8}{13} - \frac{2}{13}\log\frac{2}{13} = 1,407$$

$$I(M) = -\frac{5}{17}\log\frac{5}{17} - \frac{7}{17}\log\frac{7}{17} - \frac{5}{17}\log\frac{5}{17} = 1,565$$

Assim:

$$I(Sexo) = \frac{13}{30} * 1,407 + \frac{17}{30} * 1,567 = 1,497$$

$$G(Sexo) = 1,498 - 1,497 = 0,001$$

# Capítulo 4

# Árvore de decisão

Neste capítulo, consolidamos o que estudamos nos capítulos anteriores, implementando assim a construção elementar de uma árvore de decisão, onde, iremos utilizar simplesmente o ganho como argumento para a construção da mesma. Para isso, utilizamos dois exemplos:

## 4.1 Primeiro exemplo

Nesta secção veremos um primeiro exemplo em que usamos uma classe binária composta pelos elementos{Sim(S) e Não(N)}, correspondentes a frequência dos alunos em aulas de apoio.

Dado um conjunto $D = (x_1,\ x_2,\ x_3;\ y)$, que contem 30 observações, onde, $x_1 \in A_1 = \{F, M\};\ x_2 \in A_2 = \{[15,16] = Id_1,\ [17,18] = Id_2,\ [19,20] = Id_3\}$ e $A_3 = \{$Muito Bom(MB), Bom, Razoável(Rz), Mau$\}$, $y \in C = \{$Sim, Não$\}$.
$A_1$ representa o atributo "Sexo", $A_2$ o atributo "Idade"e $A_3$ o atributo "Qualidade das relações com os colegas".

### 4.1.1 Construção da árvore de decisão a partir do primeiro nível de partição

Para começar a construção da árvore de decisão, precisamos de calcular o ganho de cada um dos atributos, para podermos avaliar qual deles possui o melhor ganho, e desta forma, poder

determinar o primeiro nível de partição. Assim, para o atributo $A_1$, os valores são classificados como se segue:

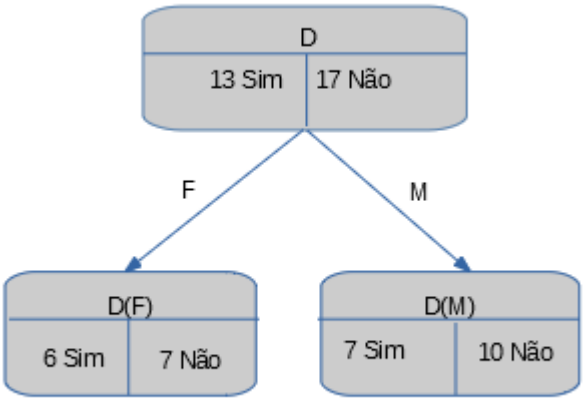



Figura 4.1: Figura das ocorrências do atributo $A_1$

Assim teremos as seguintes frequências:

Tabela 4.1: Tab.Frequências de $A_1$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | $13/30$ | $17/30$ |
| $F$ | $13/30$ | $6/13$ | $7/13$ |
| $M$ | $17/30$ | $7/17$ | $10/17$ |

onde, $q$ corresponde a probabilidade de ocorrer F ou M nas 30 observações.

Por conseguinte, calculando as impurezas teremos:

$$I(D) = -\frac{13}{30}\log_2\frac{13}{30} - \frac{17}{30}\log_2\frac{17}{30} = 0,987$$

$$I(F) = -\frac{6}{13}\log_2\frac{6}{13} - \frac{7}{13}\log_2\frac{7}{13} = 0,996$$

$$I(M) = -\frac{7}{17}\log_2\frac{7}{17} - \frac{10}{17}\log_2\frac{10}{17} = 0,997$$

$$I(A_1) = \frac{13}{30} * 0,996 + \frac{17}{30} * 0,977 = 0,986$$

Desta forma, o ganho associado ao atributo $A_1$ é:

$$G(A_1) = 0,987 - 0,986 = 0,001$$

Para o atributo $A_2$ teremos as seguintes classificações dos valores:

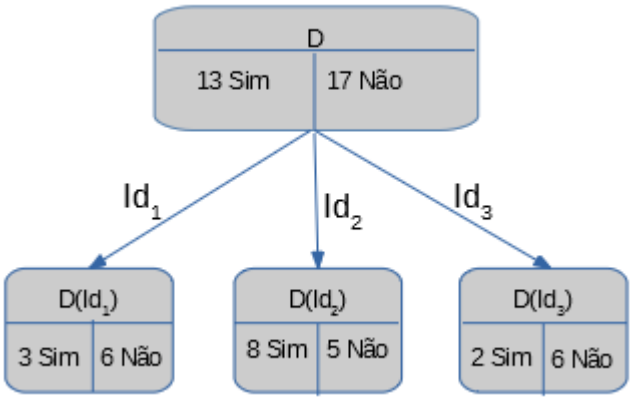



Figura 4.2: Figura das ocorrências do atributo $A_2$

As frequências de Sim e Não e as impurezas relacionadas ao conjunto $D$ serão as mesmas para os três atributos, visto que o conjunto dos dados não altera. Assim:

Tabela 4.2: Tab.Frequências de $A_2$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $Id_1$ | $3/10$ | $1/3$ | $2/3$ |
| $Id_2$ | $13/30$ | $8/13$ | $5/13$ |
| $Id_3$ | $4/15$ | $1/4$ | $3/4$ |

Efetuando o calculo das impurezas, teremos:

$$I(D) = 0,987$$

$$I(Id_1) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0,917$$

$$I(Id_2) = -\frac{8}{13}\log_2\frac{8}{13} - \frac{5}{13}\log_2\frac{5}{13} = 0,961$$

$$I(Id_3) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0,811$$

$$I(A_2) = \frac{3}{10} * 0,917 + \frac{13}{30} * 0,961 + \frac{4}{15} * 0,811 = 0,907.$$

Logo:

$$G(A_2) = 0,987 - 0,907 = 0,079$$

E para o atributo $A_3$ teremos as seguintes classificações dos valores:

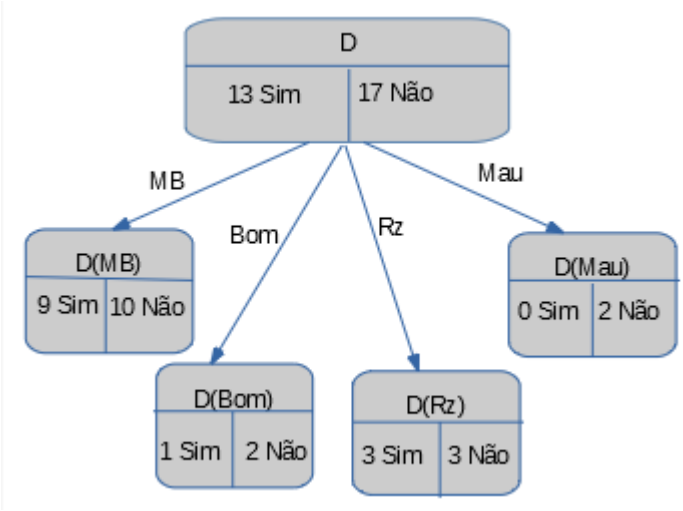



Figura 4.3: Figura das ocorrências do atributo $A_3$

Assim, teremos as seguintes frequências:

Tabela 4.3: Tab.Frequências de $A_3$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| MB | $19/30$ | $9/19$ | $10/19$ |
| Bom | $1/10$ | $1/3$ | $2/3$ |
| Rz | $1/5$ | $1/2$ | $1/2$ |
| Mau | $1/15$ | $0$ | $1$ |

Em seguida calculemos as funções de impureza:

$$I(D) = 0,987$$

$$I(MB) = -\frac{9}{19}\log_2\frac{9}{19} - \frac{10}{19}\log_2\frac{10}{19} = 0,998$$

$$I(Bom) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0,917$$

$$I(Rz) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$I(Mau) = -0\log_2 0 - 1\log_2 1 = 0$$

Assim,

$$I(A_3) = \frac{19}{30} * 0,998 + \frac{1}{10} * 0,917 + \frac{1}{5} + 0 = 0,924.$$

Logo:

$$G(A_3) = 0,987 - 0,924 = 0,063.$$

Dada a tabela 4.4,

Tabela 4.4: Tab.Ganhos para o primeiro nível de partição

| Atributos | Impurezas | Ganhos |
|---|---|---|
| $A_1$ | $0,986$ | $0,001$ |
| $A_2$ | $0,907$ | $0,079$ |
| $A_3$ | $0,924$ | $0,063$ |

podemos verificar que o atributo com o melhor ganho é o atributo $A_2$, que será considerado como a raiz da árvore de decisão, no primeiro nível de partição.

### 4.1.2 Construção da árvore a partir do segundo nível de partição

Como referenciado na secção anterior, escolhemos o atributo $A_2$ como raiz do primeiro nível de partição, por conter o melhor ganho. Assim sendo, a partir da árvore correspondente a este atributo e para cada nó contido nele, faremos o mesmo processo realizado na construção da árvore a partir do primeiro nível de partição. Ficam assim disponíveis os atributos $A_1$ $e$ $A_3$, onde vamos avaliar o ganho de cada um deles, para se poder deduzir qual o melhor atributo para continuar a construção da árvore.

Agora, vamos avaliar o ganho nos três nós do atributo $A_2$, começando pelo primeiro $Id_1$. Fazendo a classificação dos valores de $Id_1$ em relação aos atributos $A_1$ e $A_3$ temos:
Para $Id_1$ em relação a $A_1$:

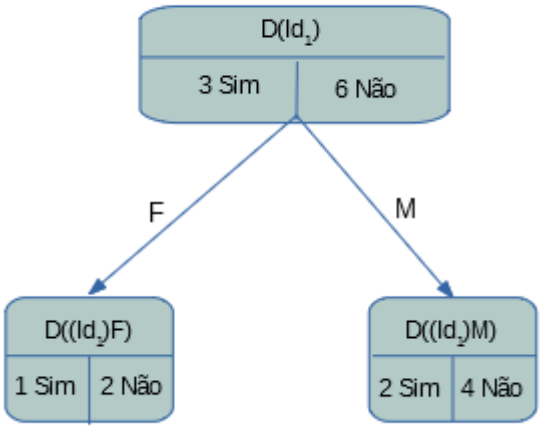



Figura 4.4: Figura das ocorrências de $Id_1$ em relação ao atributo $A_1$

Os valores correspondentes as frequências Estão indicados na tabela 4.5.

Tabela 4.5: Tab.Frequências de $Id_1$ em relação a $A_1$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | $1/3$ | $2/3$ |
| $F$ | $1/3$ | $1/3$ | $2/3$ |
| $M$ | $2/3$ | $1/3$ | $2/3$ |

Para $Id_1$ em relação a $A_3$ temos:

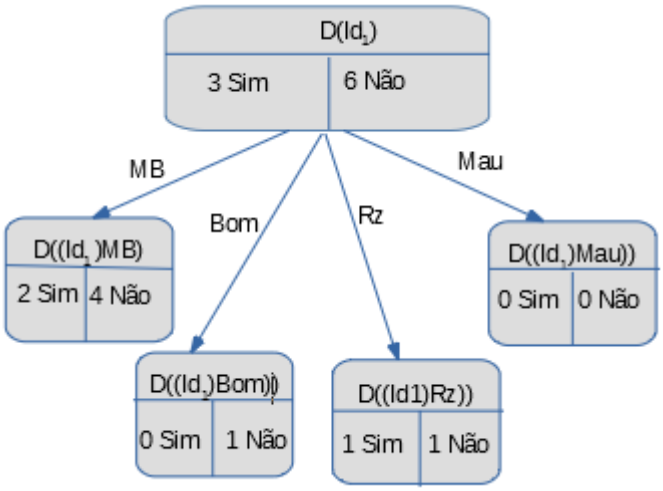



Figura 4.5: Figura das ocorrências de $Id_1$ em relação ao atributo $A_3$

Os valores correspondentes das frequências de $Id_1$ estão representados na tabela 4.6

Tabela 4.6: Tab.Frequências de $Id_1$ em relação a $A_3$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | $1/3$ | $2/3$ |
| MB | $2/3$ | $1/3$ | $2/3$ |
| Bom | $1/9$ | $0$ | $1$ |
| Rz | $2/9$ | $1/2$ | $2$ |
| Mau | $0$ | $0$ | $0$ |

Calculando o ganho deste nó para os dois atributos, teremos os resultados que estão representados na tabela a 4.7.

Tabela 4.7: Tab.Ganhos

| $Id_1$ | Impurezas | Ganhos |
|---|---|---|
| $A_1$ | $0,918$ | $0$ |
| $A_3$ | $0,834$ | $0,084$ |

Onde podemos constatar, que $Id_1$ tem o melhor ganho com o $A_3$.

Para $Id_2$ fizemos o mesmo processo que foi feito no primeiro nó. Assim sendo tivemos as seguintes classificações:
$Id_2$ em relação a $A_1$:

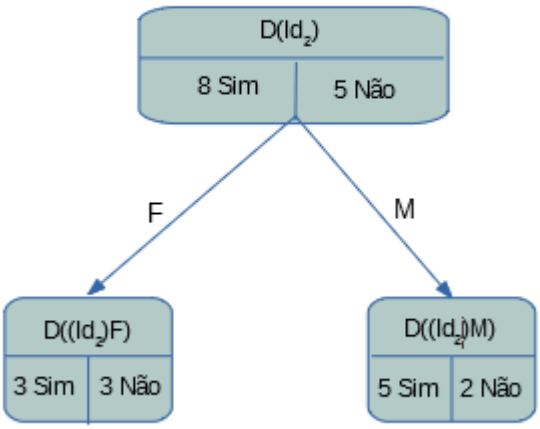



Figura 4.6: Figura das ocorrências de $Id_2$ em relação ao atributo $A_1$

Os valores correspondentes ás frequências são:

Tabela 4.8: Tab.Frequências de $Id_2$ em relação a $A_1$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | $8/13$ | $5/13$ |
| $F$ | $6/13$ | $1/2$ | $1/2$ |
| $M$ | $7/13$ | $5/7$ | $2/7$ |

$Id_2$ em relação a $A_3$:

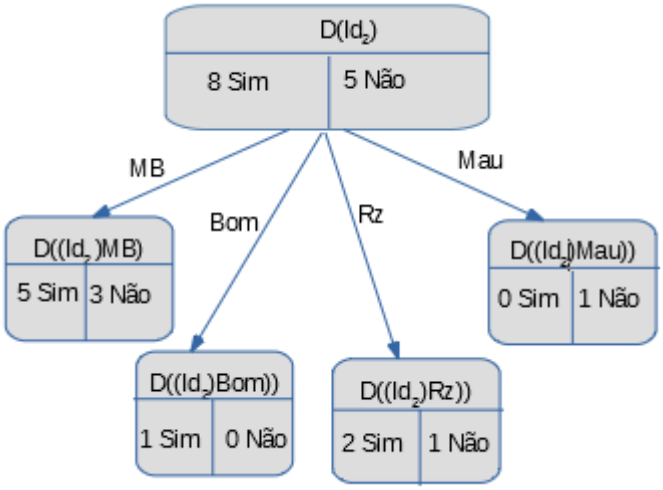



Figura 4.7: Figura das ocorrências de $Id_2$ em relação ao atributo $A_3$

Os valores correspondentes das frequências de $Id_2$ estão representados na tabela 4.9.

Tabela 4.9: Tab.Frequências de $Id_2$ em relação a $A_3$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | $8/13$ | $5/13$ |
| MB | $8/13$ | $5/8$ | $3/8$ |
| Bom | $1/13$ | $1$ | $0$ |
| Rz | $3/13$ | $2/3$ | $1/3$ |
| Mau | $1/13$ | $0$ | $1$ |

Fazendo o cálculo dos ganhos, teremos os seguintes resultados:

Tabela 4.10: Tab.Ganhos referentes a $Id_2$ em relação a $A_3$

| $Id_2$ | Impurezas | Ganhos |
|---|---|---|
| $A_1$ | $0,927$ | $0,034$ |
| $A_3$ | $0,799$ | $0,162$ |

Onde podemos constatar, que $Id_2$ têm o melhor ganho no atributo $A_3$

$Id_3$ em relação a $A_1$:

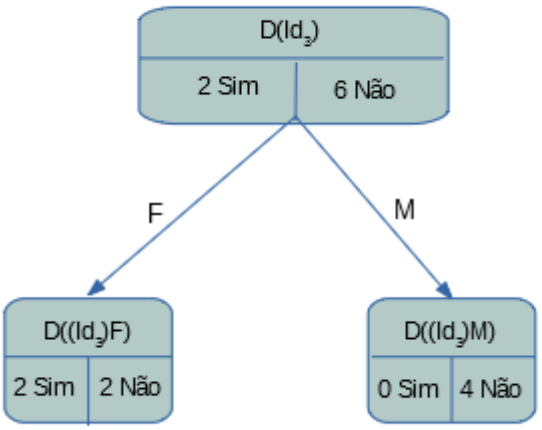



Figura 4.8: Figura das ocorrências de $Id_3$ em relação ao atributo $A_1$

Os valores correspondentes ás frequências de $Id_3$, são:

Tabela 4.11: Tab.Frequências de $Id_3$

| | q | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | $1/4$ | $3/4$ |
| $F$ | $1/2$ | $1/2$ | $1/2$ |
| $M$ | $1/2$ | $0$ | $1$ |

$Id_3$, em relação a $A_3$:

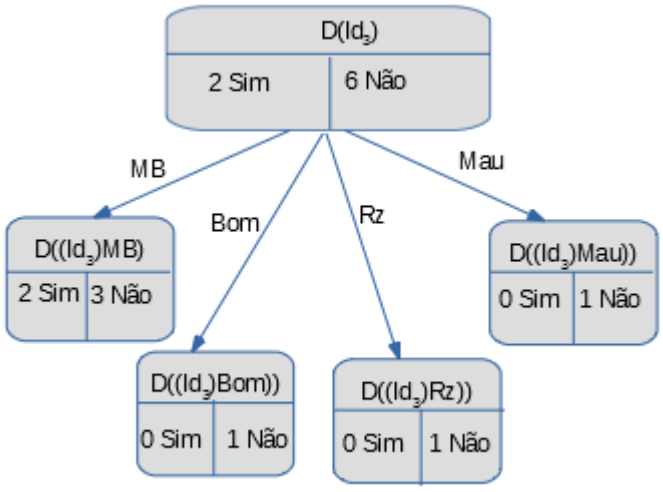



Figura 4.9: Figura das ocorrências de $Id_3$ em relação ao atributo $A_3$

Os valores correspondentes ás frequências de $Id_3$, estão representados na tabela 4.12.

Tabela 4.12: Tab.Frequências de $Id_3$em relação a$A_3$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | $1/4$ | $3/4$ |
| MB | $5/8$ | $2/5$ | $3/5$ |
| Bom | $1/8$ | $0$ | $1$ |
| Rz | $1/8$ | $0$ | $1$ |
| Mau | $1/8$ | $0$ | $1$ |

Fazendo o cálculo dos ganhos, teremos os seguintes resultados:

Tabela 4.13: Tab.Ganhos referentes a $Id_3$ em relação a $A_3$

| $Id_3$ | Impurezas | Ganhos |
|---|---|---|
| $A_1$ | $0,75$ | $0,061$ |
| $A_3$ | $0,607$ | $0,204$ |

Onde podemos constatar que $Id_3$, tem o melhor ganho também no atributo $A_3$

Assim, podemos concluir que, para os três nós, o atributo $A_3$ é o que teve o melhor ganho. No entanto, daremos continuidade à construção da árvore a partir deste atributo.

### 4.1.3 Construção da árvore a partir do terceiro nível de partição

Dando continuidade à construção da árvore, para o terceiro nível estará disponível apenas o atributo $A_1$. Portanto, não precisamos de calcular o ganho e a entropia, porque não teremos outra escolha se não a deste atributo.

Como nos casos anteriores, há a necessidade de se fazer a classificação dos valores de $A_3$ em relação a $A_1$. Assim, ao fazer a classificação dos valores, teremos dose sub-árvores, em que faremos a apresentação de apenas quatro delas referentes ao nó $Id_1$, visto que se mantém a estrutura e diferenciam-se somente os valores.

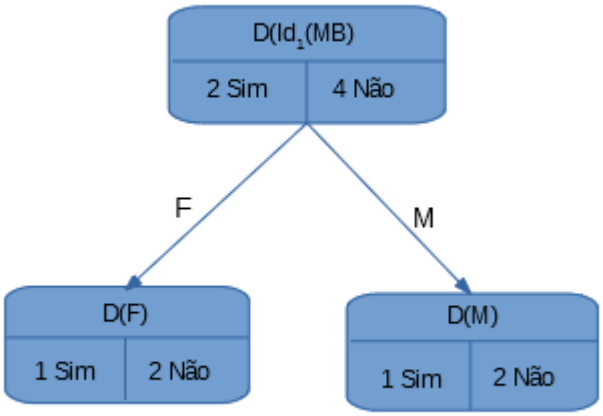



Figura 4.10: Figura das ocorrências de $MB$ em relação $A_1$ no nó $Id_1$

Para ás frequências, temos os seguintes valores:

Tabela 4.14: Tab.Frequências de $MB$ em relação a $A_1$ no nó $Id_1$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | $1/3$ | $2/3$ |
| $F$ | $1/2$ | $1/3$ | $2/3$ |
| $M$ | $1/2$ | $1/3$ | $2/3$ |

$Bom$ em relação a $A_1$ no nó $Id_1$ teremos a seguinte classificação:

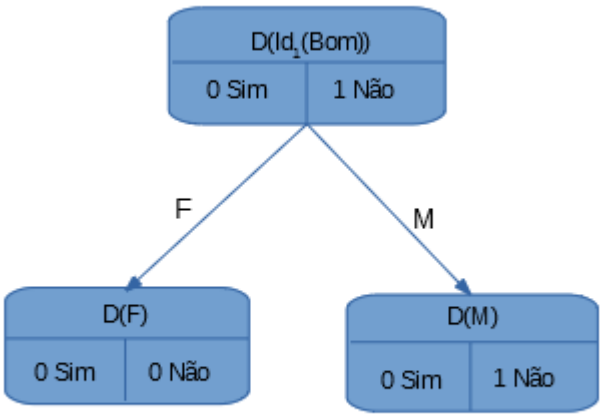



Figura 4.11: Ocorrências de $Bom$ em relação $A_1$ no nó $Id_1$

Valores das frequências:

Tabela 4.15: Tab.Frequências de $Bom$ em relação a $A_1$ no nó $Id_1$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | $0$ | $1$ |
| $F$ | $0$ | $0$ | $0$ |
| $M$ | $1$ | $0$ | $1$ |

$Rz$ em relação a $A_1$ no nó $Id_1$ teremos a seguinte classificação:

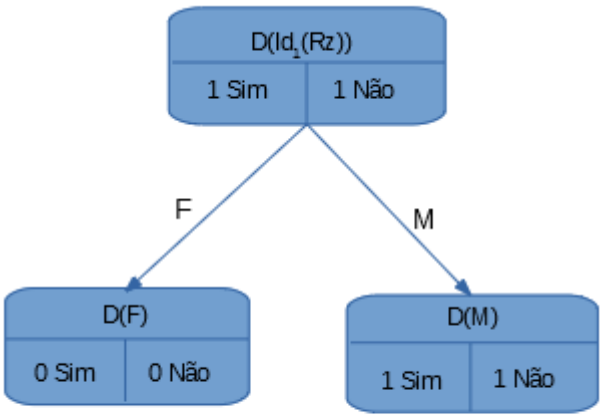



Figura 4.12: Figura das ocorrências de $Rz$ em relação a $A_1$ no nó $Id_1$

Valores das frequências:

Tabela 4.16: Tab.Frequências de $Rz$ em relação a $A_1$ no nó $Id_1$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | $1/2$ | $1/2$ |
| $F$ | $0$ | $0$ | $0$ |
| $M$ | $1$ | $1/2$ | $1/2$ |

$Mau$ em relação a $A_1$ no nó $Id_1$ temos a seguinte classificação:

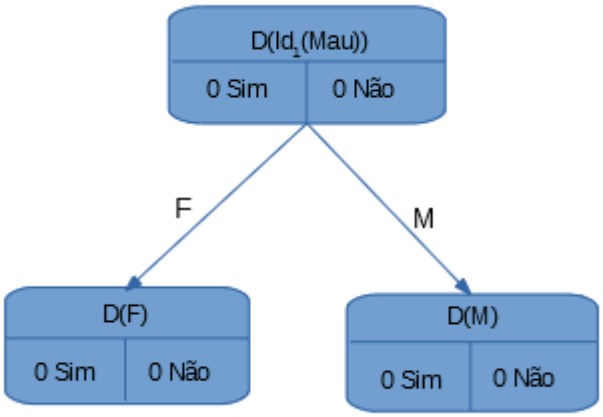



Figura 4.13: Figura das ocorrências de $Mau$ em relação a $A_1$ no nó $Id_1$

Valores das frequências:

Tabela 4.17: Tab.Frequências de $Mau$ em relação a $A_1$ no nó $Id_1$

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $D$ | | 0 | 0 |
| $F$ | 0 | 0 | 0 |
| $M$ | 0 | 0 | 0 |

Conforme foi dito na secção anterior, já não haverá continuidade da construção da árvore porque temos apenas um atributo por classificar. Assim, obtivemos a seguinte árvore de decisão:

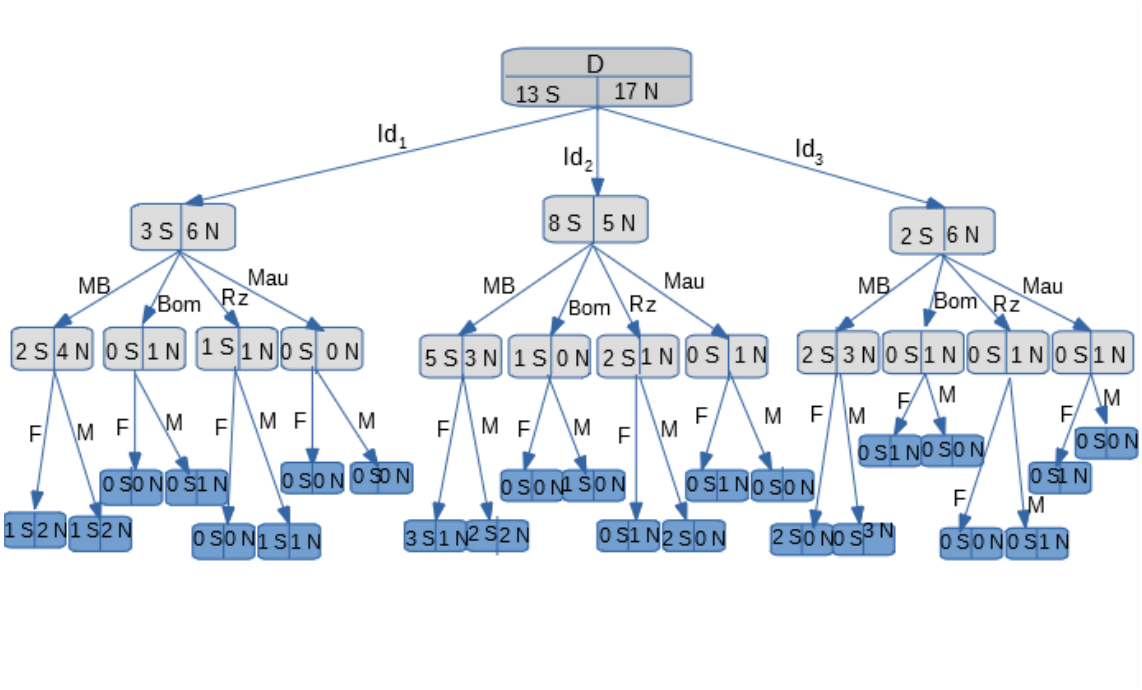



Figura 4.14: Árvore completa

Depois de construída a árvore, é preciso contabilizar o número total de nós, o número total de ramos e o número total de folhas que ela contem, pois este procedimento permite saber a dimensão da nossa árvore e, posteriormente, permite avaliar a complexidade de determinar o número de ramos ou de folhas que são necessários para fazer a predição baseada nesta árvore. Em seguida teremos os elementos caraterísticos da árvore, representados na tabela abaixo:

Tabela 4.18: Elementos característicos da árvore

| Nº de Nós | Nº de Ramos | Nº de Folhas |
|---|---|---|
| 40 | 39 | 24 |

### 4.1.4 Poda elementar da árvore

Nesta subsecção, podamos a árvore inicial, ou seja, fazemos uma primeira poda com critérios simples, que nos possibilitam eliminar alguns ramos que dão a um nó puro ou a um nó que não tenha elementos. Desta forma, estaremos perante dois critérios:

- O nó que Não tem elementos e não vale a pena continuar a particionar;
- Quando o nó é puro, não vale a pena continuar a particionar porque já temos toda a informação de que precisamos neste nó.

Assim, este processo tem, como consequência, a redução da árvore, em que teremos algumas folhas de nível dois e algumas de nível três.

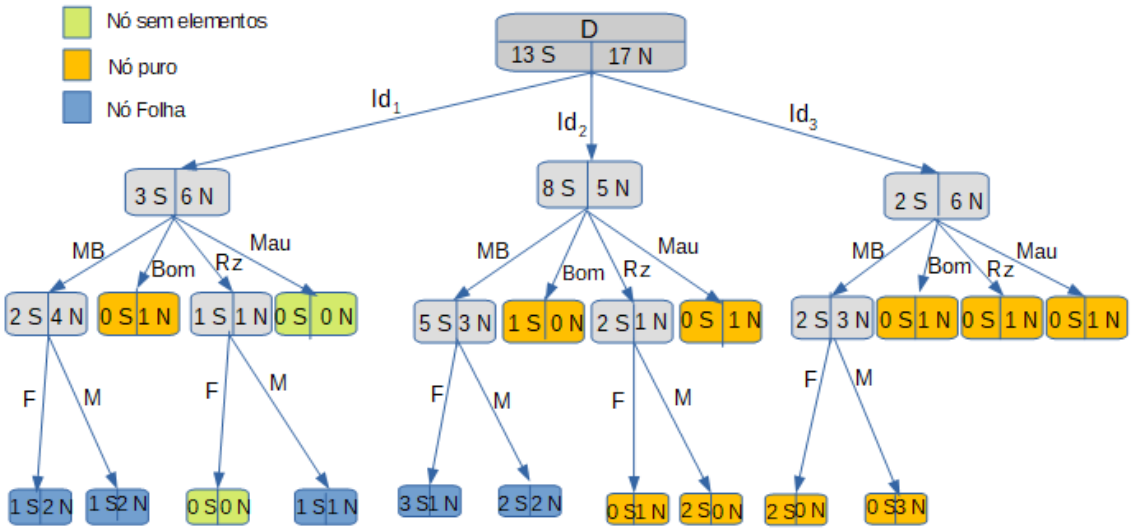



Figura 4.15: Árvore podada

Como feito anteriormente, vamos contabilizar o número total de nós, ramos e folhas da árvore podada:

Tabela 4.19: Elementos característicos da árvore

| Nº de Nós | Nº de Ramos | Nº de Folhas |
|---|---|---|
| 26 | 25 | 17 |

Para concluir este exemplo, vamos calcular o ganho total da árvore de decisão podada, a partir da diferença da entropia entre o nó inicial e todos os nós finais. Para isto, começaremos por determinar as frequências de todas as folhas:

Tabela 4.20: Frequência das folhas em relação ao ramo $Id_1$

(a)

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $MB(F)$ | $1/10$ | $1/3$ | $2/3$ |
| $MB(M)$ | $1/10$ | $1/3$ | $2/3$ |

(b)

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $Rz(F)$ | $0$ | $0$ | $0$ |
| $Rz(M)$ | $1/15$ | $1/2$ | $1/2$ |

Tabela 4.21: Frequência das folhas em relação ao ramo $Id_2$

(a)

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $MB(F)$ | $2/15$ | $3/4$ | $1/4$ |
| $MB(M)$ | $2/15$ | $1/2$ | $1/2$ |

(b)

| | $q$ | $p_1$ | $p_2$ |
|---|---|---|---|
| $Rz(F)$ | $1/30$ | $0$ | $1$ |
| $Rz(M)$ | $1/15$ | $1$ | $0$ |

Em relação ao ramo $Id_3$, as as funções de impureza serão iguais a 0. Visto que todas folhas são nós puros.

Assim, efetuando o cálculo da entropia das folhas teremos que:

$$I(Folhas) = \frac{1}{10} * 0,918 + \frac{1}{10} * 0,918 + \frac{1}{15} + \frac{2}{15} * 0,811 + \frac{2}{15} = 0,492.$$

A impureza inicial têm como valor $I(D) = 0,987$.

Logo,

$$G(Total) = 0,987 - 0,492 = 0,495$$

## 4.2 Segundo exemplo

Nesta secção faremos um Exemplo, no qual construímos a árvore de decisão da mesma maneira que construímos no exemplo anterior. A única diferença é que, ao invés de usarmos uma classe binária, usamos uma classe não binária que possui três valores.

Dado um conjunto $D = \{x_1,\ x_2,\ x_3;\ y\}$ onde, $x_1 \in A_1 = \{F, M\};\ x_2 \in A_2 = \{[15, 16] = Id_1,\ [17, 18] = Id_2,\ [19, 20] = Id_3\}$ e $x_3 \in A_3 = \{$Muito Bom(MB), Bom, Razoável(Rz), Mau$\}$. $y \in C = \{$Baixo$(B)$, Médio$(M)$, Elevado$(E)\}$.

$C$ representa a classe "Nº de faltas", onde, $B = [0, 15];\ M = [16, 27]\ e\ E = [31, 81]$.

O conjunto $D$ tem um total de 60 ocorrências, distribuídas da seguinte maneira: 34 elementos de nível baixo, 17 de nível médio e 9 de nível elevado. Logo, as probabilidades ou frequências associadas ao conjunto $D$ são: $p_B = \frac{17}{30};\ p_M = \frac{17}{60}\ e\ p_E = \frac{3}{20}$. Logo, a impureza deste que é o conjunto inicial é:

$$I(D) = -\frac{17}{30}\log_2\frac{17}{30} - \frac{17}{60}\log_2\frac{17}{60} - \frac{3}{20}\log_2\frac{3}{20} = 1,391$$

### 4.2.1 Primeiro nível de partição

Em seguida, construímos os cenários de classificação para os três atributos:

Tabela 4.22: Classificação para o atributo $A_1$

| $A_1$ | | | |
|---|---|---|---|
| | B | M | E |
| F | 16 | 8 | 3 |
| M | 18 | 9 | 6 |
| Total(D) | 34 | 17 | 9 |

Tabela 4.23: Classificação para o atributo $A_2$

| $A_2$ | | | |
|---|---|---|---|
| | B | M | E |
| $Id_1$ | 13 | 6 | 6 |
| $Id_2$ | 14 | 4 | 2 |
| $Id_3$ | 7 | 7 | 1 |
| Total(D) | 34 | 17 | 9 |

Tabela 4.24: Classificação para o atributo $A_3$

| $A_3$ | | | |
|---|---|---|---|
| | B | M | E |
| MB | 17 | 11 | 9 |
| Bom | 7 | 4 | 0 |
| Rz | 8 | 2 | 0 |
| Mau | 2 | 0 | 0 |
| Total(D) | 34 | 17 | 9 |

Em seguida, calculamos o ganho de cada atributo, para determinarmos a raiz da árvore como foi feito na secção anterior. Assim na tabela abaixo, temos o ganho e a entropia dos três atributos.

Tabela 4.25: Tab.Ganhos referentes aos três atributos

| Atributos | Impurezas | Ganhos |
|---|---|---|
| $A_1$ | $1,382$ | $0,009$ |
| $A_2$ | $1,318$ | $0,073$ |
| $A_3$ | $1,238$ | $0,249$ |

A partir da tabela 4.25, podemos constatar que o atributo que teve o melhor ganho, é o atributo $A_3$. Por conseguinte, daremos continuidade na construção da árvore a partir deste

atributo. Assim, obtém-se a figura 4.16.

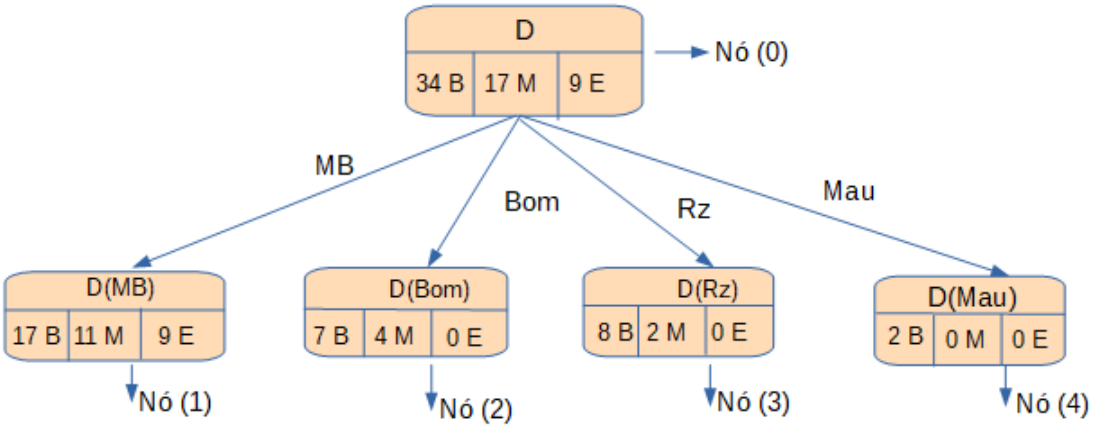



Figura 4.16: Atributo $A_3$

### 4.2.2 Segundo nível de partição

Para este nível, faremos o mesmo processo que o do primeiro nível de partição: a classificação dos nós do atributo $A_3$ em relação aos atributos $A_1$ e $A_2$.

Começamos por estudar o nó número (1):

Tabela 4.26: Classificação do nó (1) para o atributo $A_1$

| Nó (1) | | | |
|---|---|---|---|
| | B | M | E |
| F | 7 | 6 | 3 |
| M | 10 | 5 | 6 |
| Total(D) | 17 | 11 | 9 |

Tabela 4.27: Classificação do nó (1) para o atributo $A_2$

| Nó (1) | | | |
|---|---|---|---|
| | B | M | E |
| $Id_1$ | 8 | 3 | 6 |
| $Id_2$ | 6 | 3 | 1 |
| $Id_3$ | 3 | 5 | 2 |
| Total(D) | 17 | 11 | 9 |

Efetuando o cálculo da entropia e do ganho para cada uma das classificações, obtivemos os resultados apresentados na tabela abaixo.

Tabela 4.28: Tab.Ganhos Nó(1)

| Atributos | Impurezas | Ganhos |
|---|---|---|
| $A_1$ | $1,513$ | $0,018$ |
| $A_2$ | $1,457$ | $0,075$ |

Logo, verificamos que para o nó (1), o atributo com o melhor ganho é o $A_2$.

Fazendo o estudo da classificação do nó (2), obtivemos:

Tabela 4.29: Classificação do nó (2) para o atributo $A_1$

| Nó (2) | | | |
|---|---|---|---|
| | B | M | E |
| F | 4 | 2 | 0 |
| M | 3 | 2 | 0 |
| Total(D) | 7 | 4 | 0 |

Tabela 4.30: Classificação do nó (2) para o atributo $A_2$

| Nó (2) | | | |
|---|---|---|---|
| | B | M | E |
| $Id_1$ | 2 | 1 | 0 |
| $Id_2$ | 3 | 1 | 0 |
| $Id_3$ | 2 | 2 | 0 |
| Total(D) | 7 | 4 | 0 |

Efetuando os cálculos das Entropias e dos ganhos, obtivemos os seguintes resultados.

Tabela 4.31: Tab.Ganhos Nó(2)

| Atributos | Impurezas | Ganhos |
|---|---|---|
| $A_1$ | $0,942$ | $0,004$ |
| $A_2$ | $0,909$ | $0,037$ |

Logo, verificamos que para o nó (2), o atributo $A_2$ tem o melhor ganho.

Nó número (3):

Tabela 4.32: Classificação do nó (3) para o atributo $A_1$

| Nó (3) | | | |
|---|---|---|---|
| | B | M | E |
| F | 3 | 0 | 0 |
| M | 5 | 2 | 0 |
| Total(D) | 8 | 2 | 0 |

Tabela 4.33: Classificação do nó (3) para o atributo $A_2$

| Nó (3) | | | |
|---|---|---|---|
| | B | M | E |
| $Id_1$ | 3 | 2 | 0 |
| $Id_2$ | 3 | 0 | 0 |
| $Id_3$ | 2 | 0 | 0 |
| Total(D) | 7 | 2 | 0 |

Assim,

Tabela 4.34: Tab.Ganhos Nó(3)

| Atributos | Impurezas | Ganhos |
|---|---|---|
| $A_1$ | $0,604$ | $0,118$ |
| $A_2$ | $0,221$ | $0,502$ |

Desta forma, verificamos que para o nó (3), o atributo $A_2$ tem o melhor ganho.

Nó número (4):

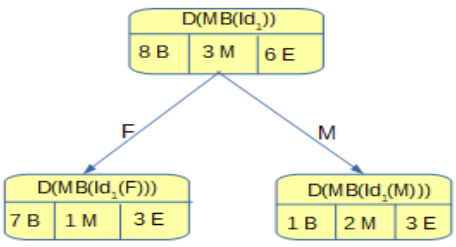
Para este nó, o ganho é igual a 0, tanto na classificação para o atributo $A_1$, como para o atributo $A_2$. Consequentemente, não há ganho neste nó e não vale a pena continuar a ramificar a árvore nele.

Assim, para este caso, chegamos á conclusão de que o atributo $A_2$ é o atributo com melhor ganho nos três nós. Logo, daremos sequência na construção da árvore a partir deste atributo.

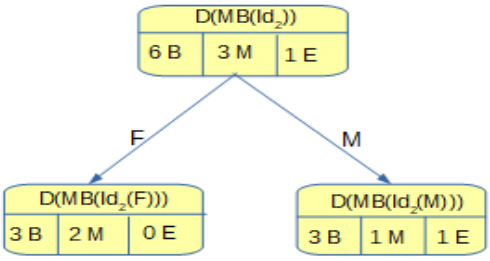
### 4.2.3 Terceiro nível de partição

Para este nível, resta apenas o atributo $A_1$, logo, não há necessidade de calcularmos o ganho, visto que não haverão outros atributos disponíveis a serem classificados. Assim, faremos a classificação dos valores de $A_2$, em relação a $A_1$.

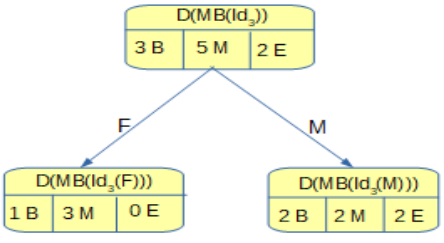
Ao fazer a classificação dos valores, teremos nove sub-árvores a partir do segundo nível de partição. Mas serão apresentadas apenas três referentes ao valor MB.




(a) MB($Id_1$)




(b) MB($Id_2$)




(c) MB($Id_3$)

Figura 4.17: Ocorrências de MB em relação ao atributo $A_1$

Tendo como frequências os seguintes valores:

| | MB | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $Id_1$ | | | | $Id_2$ | | | | $Id_3$ | | |
| | q | | p | | q | | p | | q | | p | |
| F | 11/17 | 7/11 | 1/11 | 3/11 | 1/2 | 3/5 | 2/5 | 0 | 2/5 | 1/4 | 3/4 | 0 |
| M | 6/17 | 1/6 | 1/3 | 1/2 | 1/2 | 3/5 | 1/5 | 1/5 | 3/5 | 1/3 | 1/3 | 1/3 |

Tabela 4.35

Como já não haverá continuidade da construção da árvore, visto que temos apenas um atributo por classificar. Assim, teremos a seguinte árvore de decisão:

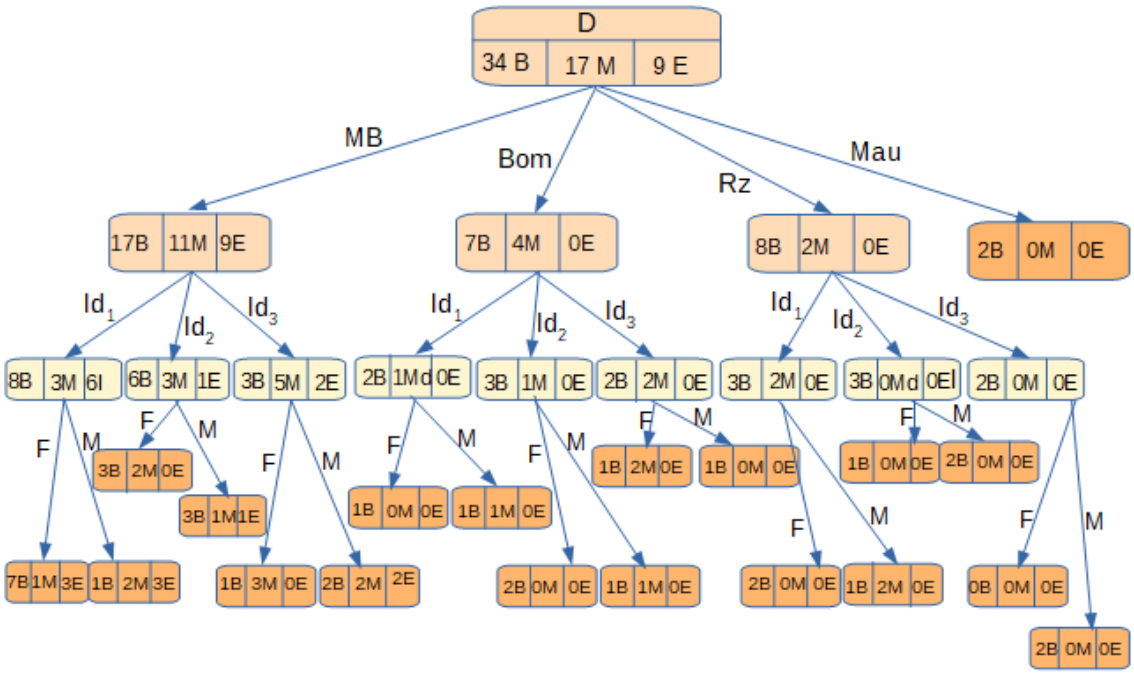



Figura 4.18: Árvore Completa2

Em seguida teremos os elementos caraterísticos da árvore, representados na tabela abaixo:

Tabela 4.36: Elementos característicos da árvore

| Nº de Nós | Nº de Ramos | Nº de Folhas |
|---|---|---|
| 32 | 31 | 19 |

### 4.2.4 Poda elementar da árvore

Nesta secção, vamos podar a árvore inicial, conforme foi explicado na secção 4.1.4 do exemplo anterior. Onde, teremos a seguinte árvore:

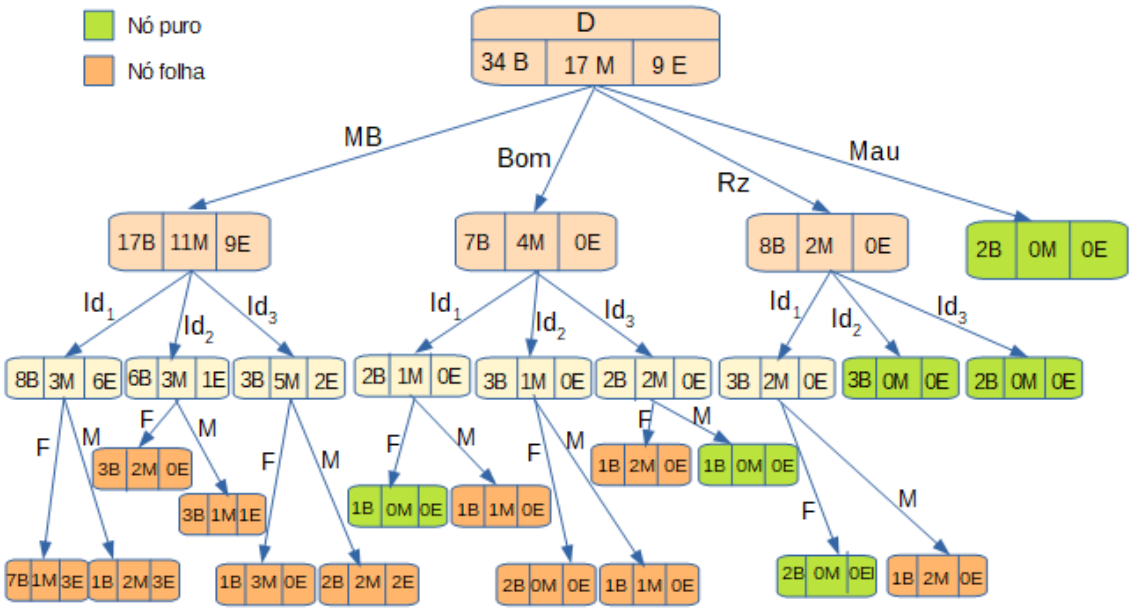



Figura 4.19: Árvore Podada2

Em seguida, contabilizamos o número total de nós, ramos e folhas da árvore podada:

Tabela 4.37: Elementos característicos da árvore Podada

| Nº de Nós | Nº de Ramos | Nº de Folhas |
|---|---|---|
| 28 | 27 | 17 |

Para terminar este exemplo, calculamos o ganho total da árvore de decisão. Começamos por determinar as frequências de todas as folhas de nível três:

Tabela 4.38: Frequência das folhas em relação ao ramo MB

(a)

| | q | p | | |
|---|---|---|---|---|
| $Id_1(F)$ | $11/60$ | $7/11$ | $1/11$ | $3/11$ |
| $Id_1(M)$ | $1/10$ | $1/6$ | $1/3$ | $1/2$ |

(b)

| | q | p | | |
|---|---|---|---|---|
| $Id_2(F)$ | $1/12$ | $3/5$ | $2/5$ | $0$ |
| $Id_2(M)$ | $1/12$ | $3/5$ | $1/5$ | $1/5$ |

(c)

| | q | p | | |
|---|---|---|---|---|
| $Id_3(F)$ | $1/15$ | $1/4$ | $3/4$ | $0$ |
| $Id_3(M)$ | $1/10$ | $1/3$ | $1/3$ | $1/3$ |

Tabela 4.39: Frequência das folhas em relação ao ramo Bom

(a)

| | q | p | | |
|---|---|---|---|---|
| $Id_1(F)$ | $1/60$ | $1$ | $0$ | $0$ |
| $Id_1(M)$ | $1/30$ | $1/2$ | $1/2$ | $0$ |

(b)

| | q | p | | |
|---|---|---|---|---|
| $Id_2(F)$ | $1/30$ | $1$ | $0$ | $0$ |
| $Id_2(M)$ | $1/30$ | $1/2$ | $1/2$ | $0$ |

(c)

| | q | p | | |
|---|---|---|---|---|
| $Id_3(F)$ | $1/20$ | $1/3$ | $2/3$ | $0$ |
| $Id_3(M)$ | $1/60$ | $1$ | $0$ | $0$ |

Para o ramo Rz, teremos apenas frequências relacionadas ao ramo $Id_1$.

Tabela 4.40: Frequência das folhas em relação ao ramo Rz

| | q | p | | |
|---|---|---|---|---|
| $Id_1(F)$ | $1/30$ | $1$ | $0$ | $0$ |
| $Id_1(M)$ | $1/20$ | $1/3$ | $2/3$ | $0$ |

Assim, fazendo o cálculo da entropia das folhas temos:

$$\begin{aligned} I(Folhas) = &\frac{11}{60} * 1,240 + \frac{1}{10} * 1,459 + \frac{1}{12} * 0,971 \\ &+ \frac{1}{12} * 1,370 + \frac{1}{15} * 0,811 + \frac{1}{10} * 1,584 \\ &+ \frac{1}{30} + \frac{1}{30} + \frac{1}{20} * 0,918 + \frac{1}{20} * 0,918 = 0,905. \end{aligned}$$

A impureza inicial têm como valor $I(D) = 1,391$.

Logo,

$$G(Total) = 1,391 - 0,905 = 0,486$$

A partir dos dois exemplos, podemos notar que a poda elementar não reduz a complexidade da árvore inicial, ou seja, a árvore continuará a ter grandes dimensões. Para resolver este problema, é preciso tentar encontrar outros critérios para que se possa fazer uma poda mais eficiente. Estes critérios serão abordados com mais detalhes nos capítulos posteriores.

# Capítulo 5

# Qualidade da Árvore

Neste capítulo, quantificamos o erro da árvore de decisão, através da matriz de confusão. Utilizaremos os dois exemplos anteriores para a construção das duas matrizes.E em seguida serão propostos indicadores para determinar a percentagem de erro.

## 5.1 Matriz de Confusão(MC)

A MC, é uma das formas que nos possibilita visualizar o desempenho de um classificador. Ela ilustra o número de predições corretas e incorrectas em cada classe. Para um determinado conjunto de dados, as linhas desta matriz representam as classes que são reais e as colunas, as classes preditas pelo classificador. Para J classes, teremos uma MC de dimensão $J \times J$.

Na tabela 5.1, podemos ver como é composta uma MC. Para este caso temos apenas duas classes:

Tabela 5.1: Esquema de uma MC(adaptada de [GCF+15])

| | S(Sim) Predição | N(Não) Predição |
|---|---|---|
| S(Sim) Real | TP(*True Positive*) | FN(*False Negative*) |
| N(Não) Real | FP(*False Positive*) | TN(*True Negative*) |

Onde:

- **TP**, refere-se aos casos em que na base de dados real, o valor é positivo e o da predição também é positivo:
- **FN**, o real é positivo, mas o predito é negativo:
- **TN**, o real é negativo e o predito também é negativo:
- **FP**, o real é negativo, mas o predito é positivo.

Assim, seja D uma base de dados, dela vamos retirar dois subconjuntos distintos. Um primeiro subconjunto que se chamará $D_{Training}$ e um segundo que se chamará $D_{Test}$. O $D_{Training}$ é o subconjunto que usaremos para construir a árvore de decisão.
Desta maneira, construímos dois tipos de MC. Uma que vai reutilizar o $D_{Training}$ e a outra que utilizará o $D_{Test}$. Por conseguinte, cada MC pode gerar um erro com naturezas diferente. Com o $D_{Training}$, encontramos o In Sample Error(ISR). Neste tipo de erro, utilizamos a MC, para comparar os dados reais com os dados da predição usando o $D_{Training}$. E para o $D_{Test}$, encontramos o Out Sample Error(OSE), que também usa a matriz de confusão para comparar os dados reais com os dados da predição, mas, usando o conjunto de dados $D_{Test}$.

## 5.2 Exemplos

Nesta secção, apresentamos dois exemplos: no primeiro exemplo usamos uma classe binária e no segundo usamos uma multi-classe com 3 valores.

### 5.2.1 Primeiro Exemplo

Em sequência, utilizamos os dados do primeiro exemplo da secção 4.1, onde temos uma base de dados composta pelos 30 primeiros eventos do conjunto D(chamamos esta base de dados de $D_{Training}$) e uma classe binária composta por {Sim(N) e Não(N)}, classificada em $13S$ e $17N$. Para o $D_{Test}$, utilizamos 39 eventos, que têm origem também do conjunto D, começando de $101 - 140$ eventos. Este subconjunto é classificado em $19S$ e $20N$. Desta forma, tendo os dados dos dois subconjuntos, podemos calcular as duas MC. A primeira será calculada com o $D_{Training}$ e a segunda com $D_{Test}$.

Assim, comparando os valores de S e N, da árvore de decisão da figura 4.15(Predição), com os do $D_{Training}$ e do $D_{Test}$, obtemos as seguintes MC:

Tabela 5.2: MC com $D_{Training}$

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 11 | 2 |
| N (Real) | 3 | 14 |

Tabela 5.3: MC com $D_{Test}$

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 7 | 12 |
| N (Real) | 6 | 14 |

Em seguida, a partir destas MC, criamos indicadores, que quantificarão o erro da árvore. Os indicadores utilizados com mais frequência são:

1. *Accuracy*, indicador usado em BD com o mesmo número de exemplos para cada classe e também, quando as penalidades de acertos e erros para cada classe forem as mesmas. A *accuracy* pode ser definida como na fórmula abaixo:

$$Accuracy = \frac{TP + TN}{n}.$$

onde, n é o número total de exemplos.

2. *Precision*, número de exemplos que são previstos pertencerem a uma classe e que realmente são dessa classe. A precisão, pode ser definida como:

$$Precision = \frac{TP}{TP + FP}$$

3. *Recall*, frequência em que o classificador encontra os exemplos numa classe. Ou seja, quando um exemplo é realmente de uma determinada classe: o quão frequente é classi-

ficado como sendo dessa classe. O *recall* pode ser definido como:

$$Recall = \frac{TP}{TP + FN}$$

4. *F1 Score*, combina a *precision* e o *recall*, de modo a trazer um número único que indique a qualidade geral do modelo de classificação. Este indicador tem melhor funcionamento em base de dados com classes desproporcionais. *O F1 Score* pode ser definido como:

$$F1Score = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

5. *Especificity*, indicador que mostra a proporção de TN, ou seja, a capacidade do classificador para dizer correctamente a ausência da condição para casos que realmente não a tem. A Especificidade pode ser definida como:

$$Especificity = \frac{TN}{TN + FP}.$$

Ao escolher um indicador, deve-se ter em conta factores como a proporção de dados de cada classe no conjunto de dados e, principalmente qual o objectivo da aplicação prática do modelo de classificação.

Para o nosso modelo, estamos a classificar o desempenho dos alunos quanto à frequência em aulas de apoio ou explicações fora da sala de aulas. O erro mais significativo para este caso, consiste no facto de o classificador prever que o aluno não frequenta aulas de apoio, mas a Base de dados dizer o contrario, ou seja, que frequenta. Assim, para este caso é mais importante minimizar os FN. Logo, podemos escolher o Recall como nosso indicador.

Para a MC do $D_{Training}$, temos:

$$Recall_{Training} = \frac{11}{11 + 2} = 0,846.$$

Para a MC do $D_{Test}$, temos:

$$Recall_{Test} = \frac{7}{7 + 12} = 0,333.$$

Ao avaliarmos o classificador por meio do *Recall*, devemos ter em conta que este terá um resultado satisfatório quando está próximo de 1. Ou seja quanto mais ele se aproximar de 0, o resultado não é satisfatório. Assim, podemos observar pelos resultados acima, que a MC da base de dados do $D_{Test}$ tem um valor muito baixo de *Recall*, isto significa que o nosso classificador não esta a funcionar como deve ser. Achamos que a razão para este mau funcionamento, se deva ao facto de não utilizarmos um número suficiente de dados para o *training*. Uma maneira de resolver este problema seria fazer o training com muito mais dados. E é isto que faremos no capítulo a seguir.
Podemos também utilizar o indicador *Specificity*, que é o complementar do *Recall*, onde minimizaremos os FP. Consequentemente, utilizando a MC do $D_{Training}$, obtivemos:

$$Specificity_{Training} = \frac{14}{14+3} = 0,824.$$

Utilizando a MC do $D_{Test}$, temos:

$$Specificity_{Test} = \frac{14}{14+6} = 0,7.$$

Assim, é possível verificar através dos dados acima, que para a *Specificity* os resultados dos dados de teste estão mais próximos de 1, do que os resultados dos dados de teste do *Recall*. O que indica que o classificador nos dados de teste está a funcionar melhor com a*Specificity*.

### 5.2.2 Segundo Exemplo

Em seguida, apresentamos o segundo exemplo, em que, utilizaremos os dados do segundo exemplo da secção 4.2.
Temos uma base de dados composta pelos primeiros 60 eventos do conjunto D($D_{Training}$), e as classes{B, M e E}, classificada em $34B$, $17M$ e $9E$. Para o $D_{Test}$, utilizamos outros 59 eventos, que também têm origem no conjunto D, mas que são diferentes do ($D_{Training}$). Este subconjunto é classificado em $31B$, $19M$ e $9E$.

Como estamos perante 3 classes, a construção da MC difere um pouco da MC com a classe binária. Neste caso, teremos uma matriz $3 \times 3$, os valores da diagonal principal constituem

o número de acertos de cada classe e os restantes valores são a quantidade de erros obtidos. Por exemplo, a Base de dados classifica como sendo da classe B, mas a árvore classifica como sendo da classe M.

Assim, teremos as seguintes MC:

| | | Classe Predita | | | Erro |
|---|---|---|---|---|---|
| | | B | M | E | |
| Classe Real | B | 24 | 5 | 4 | 9 |
| | M | 5 | 10 | 2 | 7 |
| | E | 5 | 2 | 3 | 7 |
| Erro | | 10 | 7 | 6 | 23 |

Figura 5.1: MC com $D_{Training}$ do segundo exemplo

| | | Classe Predita | | | Erro |
|---|---|---|---|---|---|
| | | B | M | E | |
| Classe Real | B | 23 | 6 | 1 | 7 |
| | M | 14 | 2 | 3 | 17 |
| | E | 7 | 3 | 0 | 10 |
| Erro | | 21 | 9 | 4 | 34 |

Figura 5.2: MC com $D_{Test}$ do segundo exemplo

Para este exemplo, classificamos o desempenho dos alunos com base ao número de faltas obtidas. O erro de confundir um E em lugar de um M é menos grave comparativamente à classificação de um E em lugar de um B. No entanto, para este caso, não classificamos o erro da mesma maneira, podemos utilizar indicadores diferentes para cada caso. Assim, um primeiro indicador que vamos utilizar é a *accuracy*. trata-se de um indicador simples, mas que pode nos ajudar a indicar se o nosso classificador está a dar um conjunto de respostas corretas. Um segundo indicador pode ser *Recall* .

Efectuando os cálculos para acuracy utilizando o $D_{Training}$, temos:

$$Accuracy_{Training} = \frac{TP}{n} = \frac{37}{60} = 0,616$$

onde n é o número total de exemplos.

Utilizando o $D_{Test}$, temos:

$$Accuracy_{Test} = \frac{25}{59} = 0,424$$

Agora vamos efectuar os cálculos para o recall, utilizando o $D_{Training}$:

$$Recall_B = \frac{24}{24+9} = 0,727$$

$$Recall_M = \frac{10}{10+7} = 0,588$$

$$Recall_E = \frac{3}{3+7} = 0,3$$

Utilizando o $D_{Test}$, temos:

$$Recall_B = \frac{23}{23+7} = 0,766$$

$$Recall_M = \frac{2}{2+17} = 0,105$$

$$Recall_E = \frac{0}{0+10} = 0$$

Podemos observar que, tanto para a *accuracy*, como para o *recall*, continuamos a ter resultados pouco satisfatórios, o que indica que o classificador não está a ter um bom funcionamento. Achamos que a razão para este mau funcionamento deve-se ao facto de não utilizarmos um número suficiente de dados da base de dados. No capítulo a seguir fazemos experiências com mais dados, a ver se minimizamos o problema.

# Capítulo 6

# Poda

Neste capítulo, abordamos a poda da árvore de decisão especificamente sobre a pré-poda, os critérios mais utilizados para a realização desta poda. Fazemos também, algumas propostas de Pré-poda e, por fim, faremos experiências utilizando a Pré-poda e, como consequência obtivemos duas árvores de decisão, uma árvore sem a Pré-poda e outra com a Pré-poda.

## 6.1 Utilização de métodos de Poda em árvores de decisão

As árvores de decisão possuem inúmeras vantagens. Apesar destas vantagens, existem também algumas desvantagens tais como em domínios com ruídos. As estratégias de poda são alguns dos métodos que permitem a redução destes ruídos. Segundo [GCF+15], dados com ruído levantam dois problemas. O primeiro é que as árvores induzidas classificam novos objectos de um modo não confiável. Os nós mais profundos refletem mais o conjunto de treino e aumentam o erro devido à variância do classificador. Este problema é conhecido como Sobre-ajuste(*Overfitting*). O segundo problema é que a árvore induzida tende a ser grande e, consequentemente, torna-se difícil compreendê-la.

Os métodos de poda podem ser realizados de duas maneiras diferentes: Pré-poda e Pós-poda. O primeiro método, que é um dos objetivos deste trabalho, consiste em parar a construção da árvore quando algum critério é satisfeito. Ou seja, este processo é feito à medida que

a árvore de decisão é gerada, antes que esta classifique perfeitamente o conjunto de dados de treino. O segundo método consiste em construir a árvore completa, e posteriormente, realizar a poda.

## 6.2 Técnicas de Pré-poda

O método de pré-poda foi introduzido juntamente com outros algoritmos para a construção de árvores de decisão, nomeadamente o C4.5 [Qui92] e CART [RM08]. Este processo, como foi dito anteriormente, impede a geração de ramos não significativos na árvore de decisão, para evitar que sejam adicionadas informações irrelevantes à mesma. Assim, esta técnica tem como principal objetivo diminuir o ruído, melhorar a qualidade de informação dos resultados, melhorar a performance, evitar o excesso de sobre-ajuste e simplificar a árvore. Desta forma, tendo uma árvore, no processo de construção estamos a analisar um nó. A questão que se levanta é se ramificamos este nó ou não. Para responder a esta questão, consideramos uma lista de critérios de pré-poda. [RM08] apresenta diversos critérios que normalmente são usados:

1. **Nível de profundidade máxima da árvore foi atingida.** Isto significa que, aquando da construção da árvore, se a profundidade atingir um valor definido como profundidade máxima, a construção cessa e são ignorados todos os dados com profundidade superior. Porém, o uso deste critério pode levar a que informação útil seja descartada sem que seja avaliada a sua relevância. Logo, seja $d$ o nível de profundidade da árvore, e $d_{max}$ a profundidade máxima definida. Se $d > d_{max}$, para-se a construção da árvore, ignorando os restantes níveis;

2. **O número de casos no nó terminal é menor que o número mínimo de casos para nó Raiz.** Com o uso deste critério, testa-se se a quantidade de informação num determinado nó é suficiente para a árvore continuar a ser construída. Assim, se um nó possuir menos do que uma determinada percentagem do total de ocorrências, a continuação de mais ramos por esse nó é desnecessária, pelo que a construção da árvore por esse nó é terminada. Assim, seja $n$ o número de elementos de um nó, $N$ número de eventos do nó raiz, e $\beta \in [0, 1]$ um parâmetro de proporção. se $n \leq \beta N$ for verdadeira, termina-se a construção de mais ramos a partir deste nó;

3. **Se o nó fosse dividido, o número de casos em um ou mais nós filhos seria menor que o número mínimo de casos para nós filhos**. Para este critério, construímos um ramo, avaliamos o número de elementos por nó, e se houver um nó que tenha mais do que uma certa percentagem dos parâmetros que estivermos a utilizar, para-se a ramificação neste nó. Assim, dada uma percentagem $\alpha \in [0, 1]$ e dado um $n_f$, que é o número de elementos em um nó filho e um $n_p$, que é o número de elementos do nó pai. Se $n_f > \alpha n_p$, para-se a ramificação da árvore através deste nó;

4. **Quando a impureza é aproximadamente 0 ou a pureza é aproximadamente 1.** Neste critério, tendo um nó D, que corresponde a uma classe E, onde $E =$ Classe(D). Se $I(E) \leq \varepsilon$, para-se o crescimento deste ramo. Logo, o nó torna-se uma folha com a classe predominante.

De acordo com estes critérios, obtém-se quatro parâmetros de poda($d_{max}$, $\beta$, $\alpha$, $\varepsilon$). Destes parâmetros há valores que correspondem a não fazer nenhuma poda, como podemos observar na tabela 6.1:

Tabela 6.1: Parâmetros de poda, quando o critério nos leva a uma poda

| Parâmetros | $d_{max}$ | $\beta$ | $\alpha$ | $\varepsilon$ |
|---|---|---|---|---|
| Descrição | Se $(d > d_{max})$ | Se $n \leq \beta N$ | Se $n_f > \alpha n_p$ | Se $I(E) \leq \varepsilon$ |
| Valores correspondentes a não haver poda | $\infty$ | 0 | 1 | 0 |

É com estes parâmetros que vamos controlar a ramificação da árvore. Assim, na secção a seguir apresentamos um exemplo de pré-poda, onde usaremos alguns destes parâmetros.

## 6.3 Aplicações usando critérios de pré-poda

Nesta secção, aplicamos uma árvore de decisão com um maior número de eventos em relação às bases de dados dos exemplos anteriores. Em primeiro lugar, construímos uma árvore de decisão, em que usaremos valores correspondentes a não realizar nenhuma poda,

em seguida fizemos mais árvores de decisão, onde, cada uma será construída com base num critério diferente de pré-poda.

Suponha-se que primeiramente se tem a seguinte base de dados, que permite testar 3 atributos ($A_1$ = Local de residência, $A_2$ = Nº de Chumbos, $A_3$ = Notas Português ) e obter a resposta à pergunta "Frequenta aulas de apoio?"

Dado um conjunto $D = \{x_1,\ x_2,\ x_3;\ y\}$, onde $x_1 \in A_1 = \{\text{Rural(R)}, \text{Urbano(U)}\};\ x_2 \in A_2 = \{\text{Dois ou mais anos(D)},\ \text{Nenhum(N)},\ \text{Um}\}$ e $x_3 \in A_3 = \{\text{Muito Bom(MB)},\ \text{Bom},\ \text{Suficiente(Suf)},\ \text{Mau}\}$; $y \in C = \{\text{Sim(S)},\ \text{Não(N)}\}$.

O conjunto $D$ tem um total de 100 ocorrências, distribuídas em 45 elementos pertencentes à classe Sim e 55 pertencentes à classe Não.

### 6.3.1 Árvore de decisão sem poda

Para esta subsecção, no processo de construção da árvore, utilizamos como parâmetros, os valores $d_{max} = \infty,\ \beta = 0,\ \alpha = 1\ e\ \varepsilon = 0$. E como consequência, com estes valores, ficam inativas quaisquer podas. Depois de construída a árvore, fizemos as duas tabelas de confusão referentes aos dados de treino e aos dados de teste, onde, por meio de indicadores, iremos avaliar a qualidade da árvore.

Assim, obtivemos a seguinte árvore de decisão e as respectivas MC:

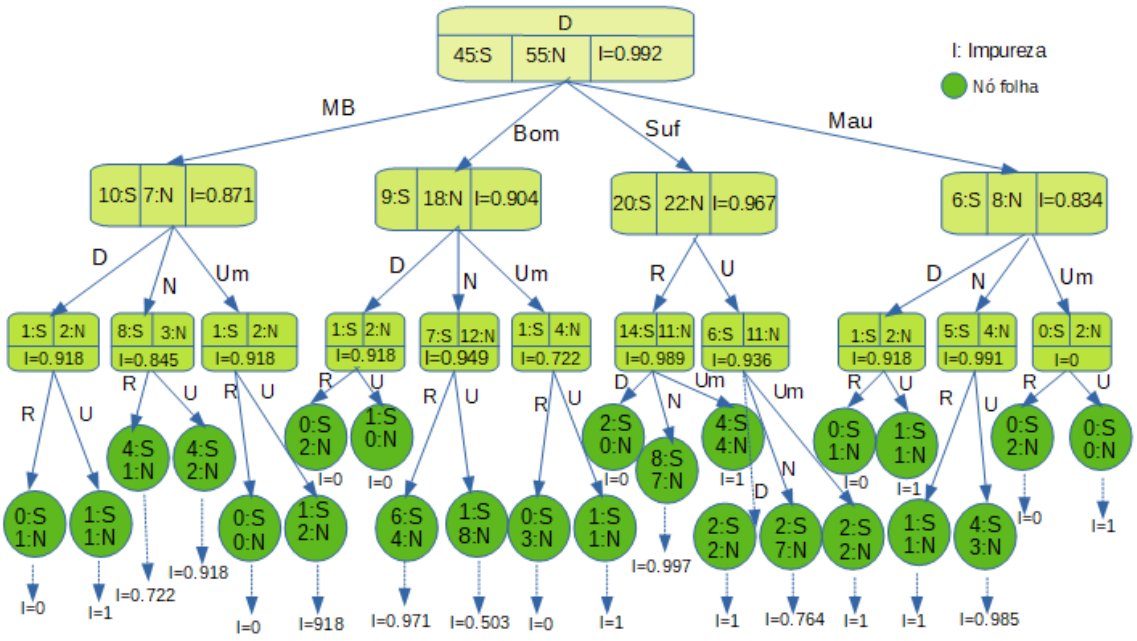



Figura 6.1: Árvore com os parâmetros sem poda

Tabela 6.2: MC com os dados de treinamento

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 34 | 12 |
| N (Real) | 23 | 31 |

Tabela 6.3: MC com os dados de teste

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 7 | 8 |
| N (Real) | 7 | 8 |

Entretanto, para o nosso modelo, minimizamos os FN, utilizando o indicador *Recall* e a sua inversa, pelas mesmas razões que as do exemplo 5.2.1. Visto que a classe é a mesma.
Para os dados de treinamento obtivemos,

$$Recall_{Training} = \frac{34}{34+12} = 0,739.$$

$$Especificity_{Training} = \frac{31}{31+23} = 0.574$$

E para os dados de teste obtivemos,

$$Recall_{Test} = \frac{7}{7+8} = 0,466.$$

$$Especificity_{Test} = \frac{8}{8+7} = 0.533$$

É de notar que os resultados são muito baixos, tanto para o *recall* dos dados de treinamento, como para o *recall* dos dados de teste. Assim, com esta experiência, estamos a ver que esta árvore, não vai funcionar bem e não vai oferecer grande possibilidade de melhoria, porque os atributos que estamos a utilizar não são muito diferenciadores. Desta maneira, os níveis de impureza continuam muito elevados mesmo ao nível das folhas. A razão para este problema deveu-se ao fato de os três atributos utilizados terem sido escolhidos aleatoriamente em um

conjunto de 49 atributos da base de dados. como consequência, nenhum deles está a ser capaz de diferenciar corretamente a informação.

Para resolver este problema, precisamos de encontrar um outro tipo de atributo na base de dados. Para isso, calculamos o ganho de impureza em todos os atributos disponíveis na base de dados e verificamos qual deles tinha o nível de impureza mais baixo. Este processo é apresentado no próximo capítulo, onde, utilizaremos uma aplicação em *python* para agilizar o processo de busca e construção da árvore de decisão.

### 6.3.2 Poda utilizando o critério de Profundidade máxima

Nesta secção, podamos á árvore de decisão, utilizando como critério da profundidade máxima, com o parâmetro $d_{max} = 2$ e, em seguida, fazemos as MC, para avaliação da qualidade da árvore. Assim obtivemos a seguinte árvore de decisão:

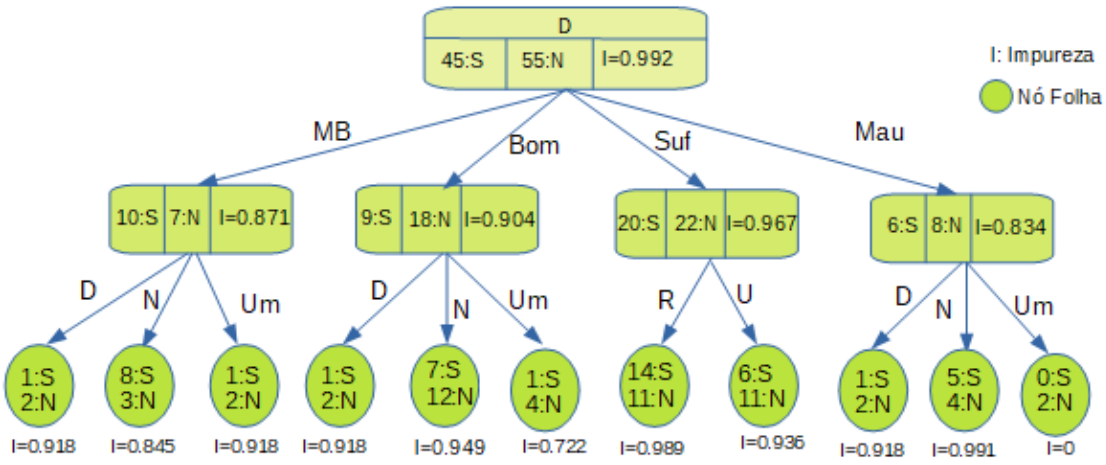



Figura 6.2: Profundidade máxima $d_{max} = 2$

Tabela 6.4: MC com os dados de treinamento, utilizando o parâmetro $d_{max}$

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 28 | 17 |
| N (Real) | 19 | 36 |

Tabela 6.5: MC com os dados de teste, utilizando o parâmetro $d_{max}$

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 7 | 9 |
| N (Real) | 6 | 8 |

Calculando o *Recall* e *Especificity* para os dados de treinamento,

$$Recall_{Training} = \frac{28}{28+17} = 0,622$$

$$Especificity_{Training} = \frac{36}{36+19} = 0.655.$$

E calculando *Recall* e *Especificity* para os dados de teste,

$$Recall_{Test} = \frac{7}{7+9} = 0,466$$

$$Especificity_{Test} = \frac{8}{8+6} = 0.571.$$

### 6.3.3 Poda utilizando o critério de percentagem mínima de elementos num nó

Nesta secção, podamos á árvore de decisão, utilizando como critério de paragem a percentagem mínima de elementos num nó, com o parâmetro $\beta = 0.10$ e, em seguida, fazemos as MC, para avaliação da qualidade da árvore. Assim obtivemos a seguinte árvore de decisão:

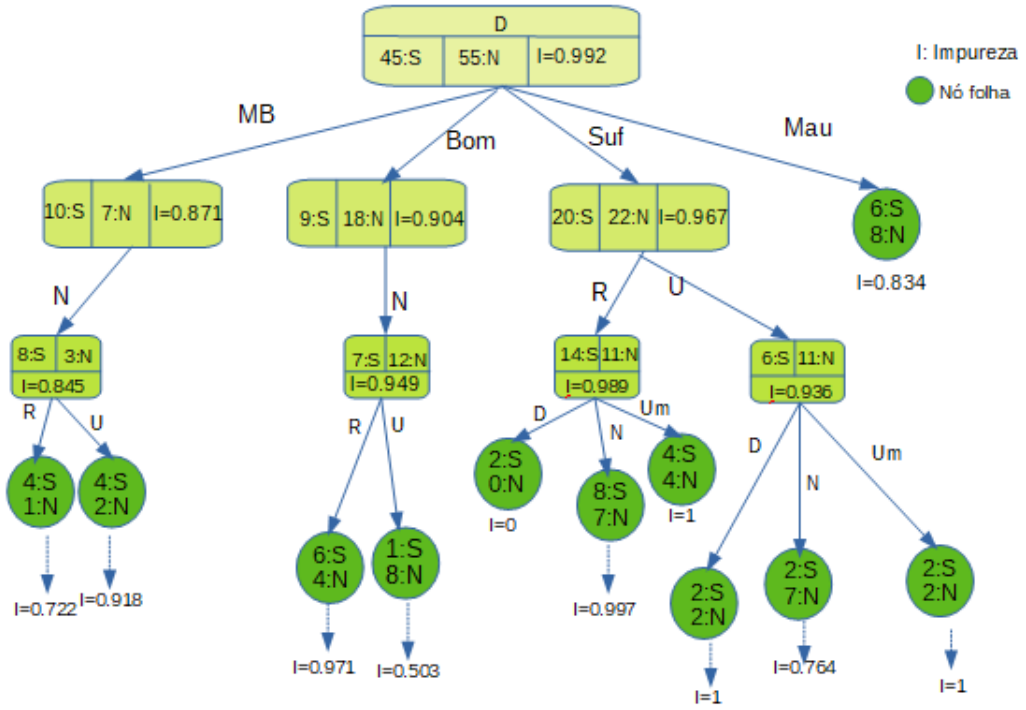



Figura 6.3: Número mínimo de elementos em um nó

Com está poda, avaliamos as consequências para os dados de treinamento e para os dados de teste, através das duas MC. E, por fim, as avaliações do *Recall* e *Especificity* para ambas MC.

Tabela 6.6: MC com os dados de treinamento, utilizando o parâmetro $\beta$

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 30 | 15 |
| N (Real) | 22 | 33 |

Tabela 6.7: MC com os dados de teste, utilizando o parâmetro $\beta$

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 4 | 11 |
| N (Real) | 7 | 8 |

Calculando o *Recall* e *Especificity* para os dados de treinamento,

$$Recall_{Training} = \frac{30}{30+15} = 0,666$$

$$Especificity_{Training} = \frac{33}{33+22} = 0.6.$$

E calculando o *Recall* e *Especificity* para os dados de teste,

$$Recall_{Test} = \frac{4}{4+11} = 0,266$$

$$Especificity_{Test} = \frac{8}{8+7} = 0.533.$$

### 6.3.4 Poda utilizando o critério de percentagem mínima em um nó filho

Nesta secção, da-se continuidade a poda da árvore, utilizando o critério de percentagem mínima de elementos em um nó filho em relação ao nó pai.

Para este critério, seria vantajoso, se os valores de $\alpha$ variassem em torno de [0.80, 0.95]. Mas, para este caso, usamos a percentagem de $\alpha = 0.60$, porque com valores acima de 0.60 não houve uma poda significativa da árvore. Assim, obtivemos a seguinte árvore e as respectivas MC:

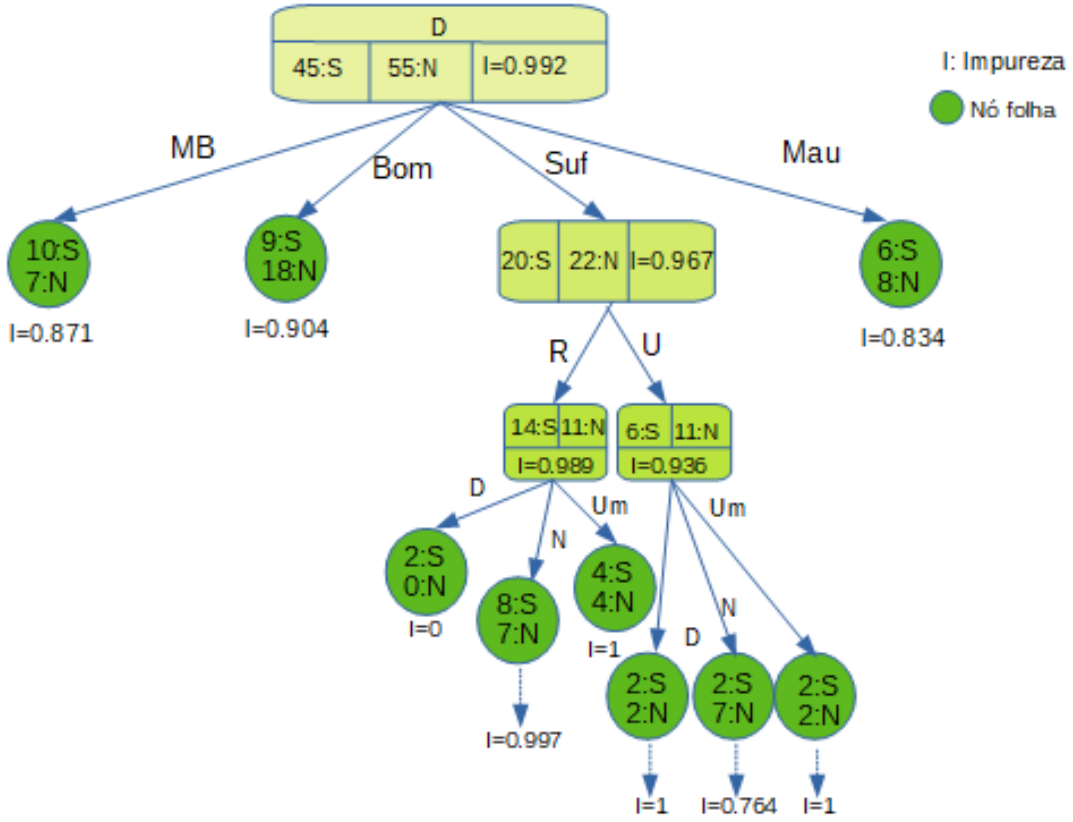



Figura 6.4: Número mínimo de elementos em um nó filho em relação ao nó pai

Tabela 6.8: MC com os dados de treinamento, utilizando o parâmetro $\alpha$

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 24 | 21 |
| N (Real) | 18 | 37 |

Tabela 6.9: MC com os dados de teste, utilizando o parâmetro $\alpha$

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 4 | 11 |
| N (Real) | 6 | 9 |

Calculando o *Recall* e *Especificity* para os dados de treinamento,

$$Recall_{Training} = \frac{24}{24+21} = 0,533$$

$$Especificity_{Training} = \frac{37}{37+18} = 0.673.$$

E calculando o *Recall* e *Especificity* para os dados de teste,

$$Recall_{Test} = \frac{4}{4+11} = 0,266$$

$$Especificity_{Test} = \frac{9}{9+6} = 0.6.$$

### 6.3.5 Poda utilizando o critério de impureza mínima

Para este critério, fizemos uma experiência de poda, com três valores do parâmetro $\varepsilon$. $\varepsilon = 0.1;\ \varepsilon = 0.3\ e\ \varepsilon = 0.5$. Em que, dos três valores, os dois primeiros eliminavam apenas nós puros e, com terceiro valor eliminou-se mais um nó que não era puro, o que ñ se pode considerar como sendo uma poda considerável. Não testamos com valores mais elevados de $\varepsilon$, porque $0.5$ é o nível de impureza mais elevado. Ou seja, acima deste valor a impureza demasiadamente grande. No entanto, obtivemos a seguinte árvore de decisão e as respectivas MC, com $\varepsilon = 0.5$:

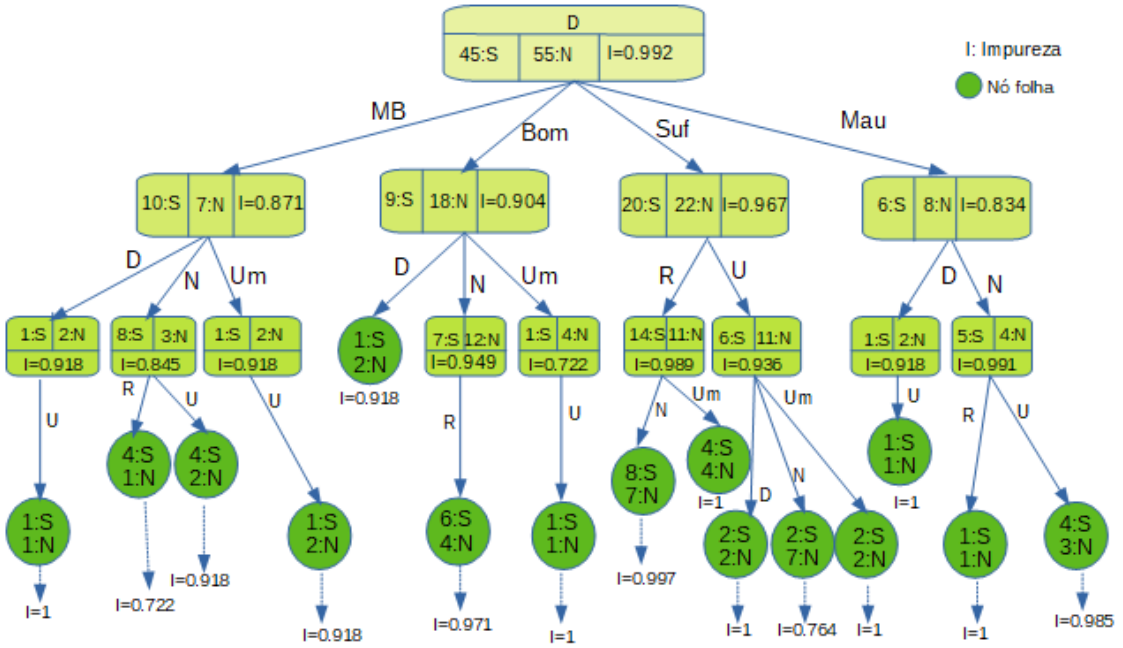



Figura 6.5: Impureza mínima

Tabela 6.10: MC com os dados de treinamento, utilizando o parâmetro $\varepsilon$

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 34 | 12 |
| N (Real) | 22 | 32 |

Tabela 6.11: MC com os dados de teste, utilizando o parâmetro $\varepsilon$

| | S (Predição) | N (Predição) |
|---|---|---|
| S (Real) | 6 | 9 |
| N (Real) | 7 | 8 |

Calculando o *Recall* e *Especificity* para os dados de treinamento,

$$Recall_{Training} = \frac{34}{34+12} = 0,739$$

$$Especificity_{Training} = \frac{32}{32+22} = 0.593.$$

E calculando o *Recall* e *Especificity* para os dados de teste,

$$Recall_{Test} = \frac{6}{6+9} = 0,4$$

$$Especificity_{Test} = \frac{8}{8+7} = 0.533.$$

O objetivo deste capítulo foi essencialmente, exemplificar a utilização da poda e dos critérios de poda. Como referenciado anteriormente, uma poda consiste em reduzir o número de ramos em uma árvore de decisão. E, para que os ramos sejam reduzidos, são necessários os critérios de poda. Cada critério foi caracterizado por um parâmetro que irá definir o nível de poda deste critério. Assim, Neste capítulo foram apresentados quatro critérios de poda, foram feitas experiências com cada critério, para vermos como eles funcionam. E obteve-se os resultados do ISE e OSR para o *Recall*, representados na tabela 6.12.

Tabela 6.12: MC com os dados de treinamento, utilizando o parâmetro $\varepsilon$

| | ISE | OSE |
|---|---|---|
| Profundidade máxima | 0.622 | 0.466 |
| Percentagem mínima de elementos | 0.666 | 0.266 |
| Percentagem mínima num nó | 0.533 | 0.266 |
| Impureza mínima | 0.739 | 0.4 |

Assim, pelos resultados apresentados na tabela 6.12 podemos observar que os critérios de Profundidade máxima e de Impureza mínima, apresentam uma taxa de erro menos elevada do que os outros dois critérios. No entanto, esperavam-se resultados mais próximos ao valor 1. Ao observamos as árvores de decisão dos critérios de profundidade máxima e de impureza mínima das figuras 6.2 e 6.5. Há mais proximidade ao tamanho da árvore de decisão sem poda a árvore do critério da impureza mínima, porque não foram eliminados muitos ramos nesta árvore. Logo para fazer a classificação podia-se usar o critério de profundidade máxima, visto que os resultados referentes a ele não se distanciam muito dos resultados do critério da

impureza mínima, principalmente em relação aos dados de teste. Ainda neste método ouve redução de mais ramos em relação á árvore do critério de impureza mínima o que indica que a árvore do critério de profundidade máxima é mais fácil de se compreender.

# Capítulo 7

# Aplicação

Neste capítulo, serão feitas experiências tendo em conta os critérios já mencionados no capítulo anterior. Porém, estes critérios de poda não serão usados apenas para exemplificar, mas servem como maneira de melhorar o funcionamento do classificador. Por sua vez, para que as experiências fossem feitas, foi desenvolvido uma *script* em *python*, recorrendo á utilização da biblioteca *scikit-learn*. Desta forma, quando é fornecido um conjunto de parâmetros referentes a um dos critérios de poda, a *script* gera e avalia duas MC referentes aos dados de teste e de treino, de acordo com os indicadores de avaliação definidos.

## 7.1 Análise dos resultados

Nesta secção, faz-se uma breve explicação da BD e a análise dos resultados por meio de três critérios de paragem.

### 7.1.1 Descrição da base de dados

Para os testes da aplicação, recorreu-se a uma BD, referente a escola secundária Conde de Monsaraz, a qual pertence ao agrupamento vertical de Escolas de Reguengos de Monsaraz. No ano lectivo de 2017/2018, esta escola contou com:

- Seis Turmas de 10º ano, o que perfaz um total de 156 alunos.
- Cinco Turmas de 11º ano, o que perfaz um total de 112 alunos.

- Quatro Turmas de 12º ano, o que perfaz um total de 97 alunos.

Assim, no total a escola contou com 365 alunos distribuídos pelos 10º, 11º e 12º anos.

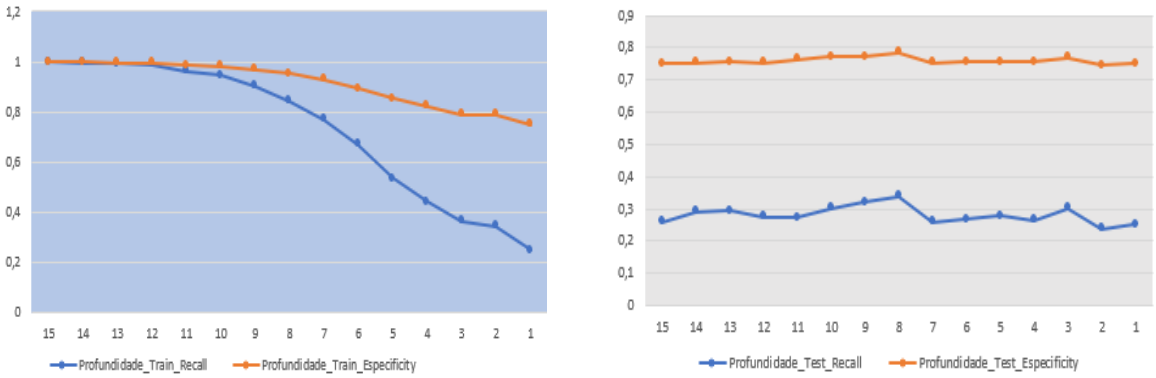
A BD que utilizamos foi modificada, para permitir agrupar um conjunto de valores de modo que possam ser adequados para o código. Assim, foram feitas duas mudanças na BD original. Primeiramente as classificações das notas de português foram calculadas mediante uma média e agrupou-se estes valores em quatro categorias que são, (Muito Bom, Bom, Suficiente e Mau). Foi modificado também o atributo Nº de faltas com as seguintes classificações: Baixo, Médio e Elevado. A BD modificada é composta por 47 colunas das quais, as primeiras 46 são os atributos, e a ultima coluna serve de classe. Assim, a árvore de decisão será criada na base desta BD. No entanto, a BD foi modificada para avaliar o desempenho dos alunos com base num conjunto de atributos, classificados mediante as notas de português.

Primeiramente, é necessário fazer a construção da árvore com parâmetros que não incitam a poda. Esta árvore será usada como referência para comparar com os resultados obtidos no decorrer da aplicação dos diversos critérios de paragem. Assim, o conjunto de treino é composto pelos primeiros 80% dos dados que compõem a BD, sendo os restantes 20% usados como teste. Assim, o resultado será representado por intermédio de duas MC, uma referente aos dados de treino e a outra referente aos dados de teste. A partir das duas serão aplicados dois indicadores, o *Recall* e a *Especificity*, que por sua vez irão avaliar a eficiência do método de Pré-poda para permitir diferenciar se houve ou não melhorias com o uso de tal critério de poda.

Nas secções seguintes, são testados alguns dos critérios acima apresentados de forma a analisar a sua influência na construção da árvore de decisão. Espera-se que através desta análise se possa compreender as melhorias obtidas pela utilização do método em questão.

### 7.1.2 Análise dos resultados em relação ao parâmetro de profundidade

Para este teste, usou-se apenas o nível de profundidade máxima. Executando o código com diferentes valores de profundidade da árvore, obtiveram-se os seguintes resultados:



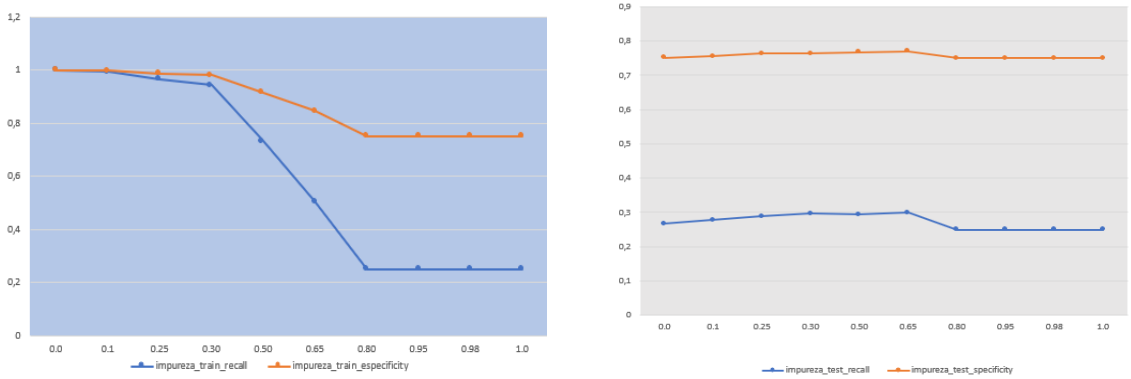

(a) Dados de treinamento (b) Dados de teste

Figura 7.1: Gráficos com ISE e OSE da profundidade

Os valores obtidos demonstram que para os dados de treinamento não ouve perda de informação em níveis superiores a 15. Tendo assim valores de *Recall* e *Especificity* iguais a 1. No entanto para níveis inferiores a 15, verifica-se a perda de informação e consequentemente o *Recall* e a *Especificity* começam a diminuir consideravelmente. Para os dados de teste temos perdas consideráveis de informação. É de notar no gráfico que os valores do *Recall* e da *Especificity* são muito próximos um do outro, o que indica que na MC os valores de TP e TN não são preponderantes em relação aos FN e FP.

### 7.1.3 Análise dos resultados em relação ao parâmetro de impureza

Seguidamente testou-se a influência da impureza mínima existente em cada nó na construção da árvore. Os resultados obtidos são aqueles que se encontram na figura 7.2:



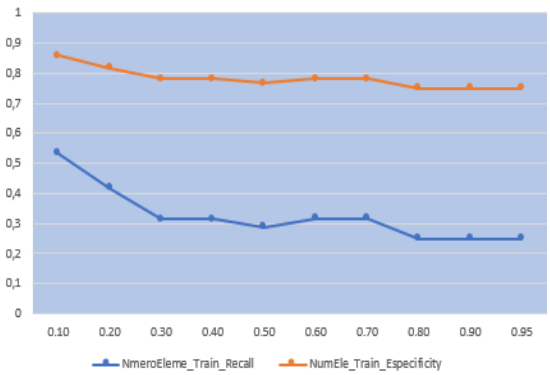

(a) Dados de treinamento (b) Dados de teste

Figura 7.2: Gráficos com ISE e OSE da Impureza

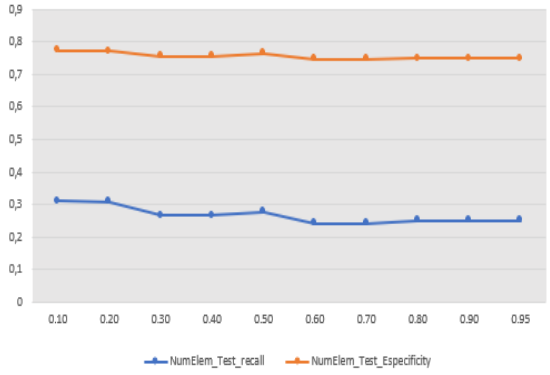
Por intermédio do gráfico, podemos verificar que nos dados de treinamento quando estamos perante uma impureza mínima igual a 0, o que implica que estamos perante uma classe pura, os valores do *Recall* e a *Especificity* são iguais a 1. Assim, no intervalo de 10% a 65% houve perda de informação. Esta perda foi aumentando à medida que as percentagens de impureza foram crescendo. Para os dados de teste temos um cenário que não difere muito do que tivemos no critério anterior tendo os valores do *Recall* e a *Especificity* muito baixos e próximos uns dos outros. No entanto, a partir da percentagem de 80% não houve alteração nos valores dos indicadores.

### 7.1.4 Análise dos resultados em relação ao parâmetro de percentagem mínima de elementos num nó

O ultimo teste baseia-se na definição de percentagens mínima de elementos que compõe um nó, de forma a determinar se a quantidade de informação existente é suficiente para continuar a ramificação da árvore nesse nó. Assim, executando vários testes com diferentes valores obtém-se os seguintes resultados:




(a) Dados de treinamento




(b) Dados de teste

Figura 7.3: Gráficos com ISE e OSE de Número mínimo de elementos

Neste teste observa-se que os valores máximos de percentagem do *Recall* e a *Especificity* ocorrem em 10%. Isto acontece para os dados de treinamento e para os dados de teste.

Assim, pelos resultados obtidos, pode-se observar que nenhum dos critérios usados nos testes estão a oferecer grandes melhorias ao classificador. No entanto os dados de treinamento

em algum momento funcionam bem. Mas, no momento em que se utilizam dados que não pertencem a este conjunto, já não observamos um bom funcionamento. Chegando-se assim a conclusão de que os dados recolhidos não permitem fazer uma correlação dos atributos e os resultados. Ou seja, com base nos atributos utilizados não se consegue determinar qual vai ser a classificação do aluno.

# Capítulo 8

# Conclusões

Com este trabalho foi possível compreender e aprofundar os conhecimentos obtidos acerca da ECD, precisamente o método de classificação aplicado à construção de árvores de decisão e do método de Pré-poda. Foi possível a compreensão de certos conceitos sobre teoria da informação, nomeadamente as frequências ou probabilidades, e de como estas são definidas através da informação referente à correlação entre os atributos e as classes.

Podemos ver também que a impureza, avalia a quantidade de informação que se perde quando passamos os dados de uma visão microscópica para uma visão macroscópica. Desta maneira, o ganho de informação de um determinado atributo é calculado por meio de funções de impureza.

Ademais, por meio do ganho de informação podemos construir uma árvore de decisão, começando por avaliar qual dos atributos obteve o melhor ganho, decidindo quem é a raiz da árvore e, assim por diante, até atingirmos o último nível da árvore. Desta maneira, faz-se uma poda elementar da árvore com critérios simples o que não reduz em grande escala a complexidade da árvore.

No capítulo 5 foram feitas várias experiências sucintas de validação com um número reduzido de elementos da BD e, como o classificador não apresentava bons resultados, deduziu-se que o problema deveu-se à falta de mais dados.

Neste trabalho também foi possível compreender conceitos ligados ao método de pré-poda bem como os diferentes critérios de paragem utilizados no método em questão. E, mais uma uma vez, testamos o método de pré-poda com os diferentes critérios de paragem e os resultados continuavam a não ser satisfatórios. Desta vez deduziu-se que a escolha aleatória dos atributos em um conjunto de 47 da BD não foi satisfatória, visto que os três atributos não estavam a ser diferenciadores. Com isto tem de se escolher entre todos os atributos da BD qual é que está a reduzir o nível de impureza.

Por fim, através da realização de experiências, testou-se a eficácia dos critérios de paragem do método de pré-poda para uma base de dados aplicada ao ensino. Podendo-se assim concluir que nenhum dos critérios utilizados apresentava melhorias para o classificador, devido aos atributos não apresentarem correlação com os resultados. Assim, com extensão a este trabalho, seria necessário considerar a constituição de uma nova BD mais adequada para indicar os atributos implícitos ao sucesso do aluno na escola.

# Bibliografia


[EMS02] Zied Elouedi, Khaled Mellouli, and Philippe Smets. A pre-pruning method in belief decision trees. In *The Ninth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU*, volume 1, pages 579–586, 2002.

[GCF+15] João Gama, André Carlos Ponce de Leon Carvalho, Katti Faceli, Ana Carolina Lorena, Márcia Oliveira, et al. Extracção de conhecimento de dados: data mining. *Available in the net*, 2015.

[JM15] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[Loh11] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[Mar07] João Maroco. *Análise estatística com utilização do SPSS*. Edições Sílabo, 2007.

[McC98] RB McCall. Fundamental statistics for behavioral sciences, brooks, 1998.

[PF91] Gregory Piateski and William Frawley. *Knowledge discovery in databases*. MIT press, 1991.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[Qui87] JR Quinlan. Simplifying decision trees. int j human-computer studies. *51 (2)*, pages 497–510, 1987.

[Qui92] JR Quinlan. C4. 5: Programs for machine learning. los atos morgan kaufmann. *Available in the net*, 1992.

[RM05] Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005.

[RM08] Lior Rokach and Oded Z Maimon. *Data mining with decision trees: theory and applications*, volume 69. World scientific, 2008.

[Wir85] N Wirth. Algorithms and data structures. oberon version: August 2004. *Available in the net*, 1985.

# Capítulo 9

# Anexos

## Script em Python

```
%import numpy as np
#!python

# Run this program on your local python
# interpreter, provided you have installed
# the required libraries.

# Importing the required packages
import numpy as np
import pandas as pd
import graphviz
from sklearn import tree
from sklearn import preprocessing
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

# Function importing Dataset
```

```
21 def importdata():
22     #balance_data = pd.read_csv('https://archive.ics.uci.edu/ml/
           machine-learning-databases/balance-scale/balance-scale.data',
           sep= ',', header = None)
23     #balance_data = pd.read_csv('./student-small.csv')
24     balance_data = pd.read_csv('./Dados_Filtrados.csv')
25     # Printing the dataswet shape
26     print ("Dataset␣Lenght:␣", len(balance_data))
27     print ("Dataset␣Shape:␣", balance_data.shape)
28     # Printing the dataset observations
29     #print ("Dataset: ",balance_data.head(1))
30     #print(balance_data.axes)
31     return balance_data
32
33 # Function to split the dataset
34 def splitdataset(balance_data):
35     begin_x=1;end_x=46;begin_y=46;end_y=47
36     index=balance_data.axes
37     X_features=np.asarray(index[1][begin_x:end_x])
38     Y_features=np.asarray(index[1][begin_y:end_y])
39     #print("-----")
40     #print(X_features,Y_features)
41     # Separating the target variable
42     features=balance_data.to_numpy(dtype=str)
43     X = features[:, begin_x:end_x]
44     Y = features[:, begin_y:end_y]
45     #print("-----")
46     #print(X,Y)
47     #print("-----")
48     # Spliting the dataset into train and test
49     X_train, X_test, y_train, y_test = train_test_split(X, Y,
           test_size = 0.3, random_state = 100)
```

```
50      return X, Y, X_train, X_test, y_train, y_test,X_features,
            Y_features
51
52  # Function to encode data
53  def encodedataset(X,leX):
54      Xencode=[];
55      for row in X:
56          Xencode.append(leX.transform(row))
57      return Xencode
58
59  # Function to perform training with giniIndex.
60  def train_using_gini(X_train,Y_train,d,epsi,alpha):
61      alpha=alpha+1e-5
62      clf_gini = DecisionTreeClassifier(criterion = "gini",
            min_impurity_split=epsi,
63                                        max_depth=d, min_samples_split=
                                              alpha)
64      clf_gini.fit(X_train, Y_train,check_input=True)
65      return clf_gini
66
67  # Function to perform training with entropy.
68  def train_using_entropy(X_train,Y_train,d,epsi,alpha):
69      alpha=alpha+1e-5
70      clf_entropy = DecisionTreeClassifier(criterion = "entropy",
            min_impurity_split=epsi,
71                                           max_depth=d,
                                                 min_samples_split=alpha)
72      clf_entropy.fit(X_train,Y_train)
73      return clf_entropy
74
75
76  # Function to make predictions
77  def prediction(X_test, clf_object):
```

```
78      Y_pred = clf_object.predict(X_test)
79      return Y_pred
80
81  # Function to calculate accuracy
82  def ConfusionMatrix(Y_test, Y_pred,leY):
83      nn=(leY.classes_).size
84      CM=np.zeros([nn,nn])
85      for i in range(0,len(Y_test)-1):
86          CM[Y_test[i],Y_pred[i]]+=1
87      return CM
88
89  # Driver code
90  def main():
91
92      # Building Phase
93      data = importdata()
94      X, Y, X_train, X_test, Y_train, Y_test,X_features,Y_features =
            splitdataset(data)
95      leX = preprocessing.LabelEncoder()
96      leY = preprocessing.LabelEncoder()
97      leX.fit(X.flatten())
98      leY.fit(Y.flatten())
99      #print(leX.classes_ ,leY.classes_)
100     Xencode=encodedataset(X,leX)
101     Xencode_train=encodedataset(X_train,leX)
102     Xencode_test=encodedataset(X_test,leX)
103     Yencode=encodedataset(Y,leY)
104     Yencode_train=encodedataset(Y_train,leY)
105     Yencode_test=encodedataset(Y_test,leY)
106
107
108     #print(Xencode[2])
109     #print(leX.inverse_transform(Xencode[2]))
```

```
110     #d=25;epsi=0.0;alpha=0.05;
111     d = int(input("profundidade maxima:"))
112     epsi=float(input("impuresa minima:"))
113     alpha=float(input("racio de popula ao minima nos:"))
114 #---------  with gini index ----------
115     clf_gini = train_using_gini(Xencode_train,Yencode_train,d,epsi,
            alpha)
116     print("Training with Gini:")
117     Yencode_pred_gini_train = prediction(Xencode_train, clf_gini)
118     CM_train=ConfusionMatrix(Yencode_train, Yencode_pred_gini_train,
            leY)
119     print("Confusion Matrix train");print(CM_train)
120     Yencode_pred_gini_test = prediction(Xencode_test, clf_gini)
121     CM_test=ConfusionMatrix(Yencode_test, Yencode_pred_gini_test,leY)
122     print("Confusion Matrix test");print(CM_test)
123     dot_data = tree.export_graphviz(clf_gini, out_file=None,filled=
            True, rounded=True,special_characters=True)
124     graph = graphviz.Source(dot_data)
125     graph.render("gini")
126 #---------  with entropy ----------
127     '''
128     clf_entropy = train_using_entropy(Xencode_train,Yencode_train,d,
            epsi,alpha)
129     print("Training with Entropy:")
130     Yencode_pred_entropy_train = prediction(Xencode_train,
            clf_entropy)
131     CM_train=ConfusionMatrix(Yencode_train,
            Yencode_pred_entropy_train,leY)
132     print("Confusion Matrix train");print(CM_train)
133     Yencode_pred_entropy_test = prediction(Xencode_test, clf_entropy)
134     CM_test=ConfusionMatrix(Yencode_test, Yencode_pred_entropy_test,
            leY)
135     print("Confusion Matrix test");print(CM_test)
```

```
136     dot_data = tree.export_graphviz(clf_entropy, out_file=None, filled
            =True, rounded=True, special_characters=True)
137     graph = graphviz.Source(dot_data)
138     graph.render("entropy")
139     '''
140 # Calling main function
141 if __name__=="__main__":
142     main()
```

# Base de dados

| Aluno | Local/residencia | Chumbos | Nport | Apoio |
|---|---|---|---|---|
| 1 | Rural | Um | MB | Sim |
| 2 | Rural | Nenhum | Suf | Não |
| 3 | Rural | Nenhum | Mau | Sim |
| 4 | Urbano | Um | Mau | Não |
| 5 | Urbano | Nenhum | MB | Sim |
| 6 | Rural | Um | MB | Não |
| 7 | Urbano | Dois ou mais | Suf | Sim |
| 8 | Urbano | Um | Bom | Não |
| 9 | Rural | Um | Mau | Sim |
| 10 | Urbano | Dois ou mais | MB | Não |
| 11 | Rural | Um | Suf | Sim |
| 12 | Urbano | Nenhum | Mau | Sim |
| 13 | Urbano | Nenhum | Mau | Não |
| 14 | Urbano | Nenhum | Bom | Não |
| 15 | Urbano | Nenhum | Bom | Sim |
| 16 | Urbano | Nenhum | Mau | Sim |
| 17 | Urbano | Dois ou mais | Bom | Sim |
| 18 | Urbano | Nenhum | Mau | Não |
| 19 | Rural | Dois ou mais | Suf | Não |
| 20 | Urbano | Nenhum | Suf | Não |
| 21 | Urbano | Dois ou mais | Suf | Não |
| 22 | Urbano | Nenhum | Suf | Sim |
| 23 | Urbano | Nenhum | Suf | Sim |
| 24 | Rural | Dois ou mais | Mau | Sim |
| 25 | Rural | Nenhum | Suf | Não |
| 26 | Urbano | Um | Suf | Sim |
| 27 | Rural | Dois ou mais | Mau | Não |
| 28 | Rural | Nenhum | Suf | Não |
| 29 | Rural | Nenhum | Bom | Não |
| 30 | Rural | Um | Suf | Sim |

| Aluno | Local de residência | Nº de chumbos | Notas Português 3º Período | Fequenta Apoio\explicações |
|---|---|---|---|---|
| 1 | Urbano | Um | MB | Sim |
| 2 | Rural | Nenhum | MB | Sim |
| 3 | Rural | Nenhum | Suf | Não |
| 4 | Urbano | Nenhum | Suf | Não |
| 5 | Urbano | Nenhum | Bom | Não |
| 6 | Rural | Nenhum | Bom | Não |
| 7 | Urbano | Dois ou mais | MB | Sim |
| 8 | Rural | Dois ou mais | Suf | Sim |
| 9 | Urbano | Nenhum | Mau | Sim |
| 10 | Rural | Um | Suf | Sim |
| 11 | Rural | Um | Mau | Não |
| 12 | Urbano | Nenhum | Bom | Não |
| 13 | Urbano | Nenhum | Suf | Não |
| 14 | Urbano | Um | MB | Não |
| 15 | Rural | Nenhum | Suf | Sim |
| 16 | Urbano | Nenhum | Bom | Não |
| 17 | Rural | Um | Mau | Não |
| 18 | Rural | Nenhum | MB | Não |
| 19 | Urbano | Nenhum | Suf | Não |
| 20 | Rural | Nenhum | Suf | Sim |
| 21 | Urbano | Nenhum | MB | Não |
| 22 | Urbano | Dois ou mais | Mau | Sim |
| 23 | Urbano | Nenhum | Suf | Sim |
| 24 | Rural | Nenhum | Suf | Sim |
| 25 | Rural | Um | Bom | Não |
| 26 | Rural | Nenhum | Bom | Sim |
| 27 | Rural | Nenhum | Bom | Não |
| 28 | Rural | Nenhum | Suf | Não |
| 29 | Rural | Nenhum | Bom | Sim |
| 30 | Urbano | Um | Suf | Não |
| 31 | Rural | Nenhum | MB | Sim |
| 32 | Urbano | Nenhum | Mau | Não |
| 33 | Urbano | Nenhum | MB | Sim |
| 34 | Urbano | Dois ou mais | Bom | Sim |
| 35 | Rural | Nenhum | Bom | Não |
| 36 | Rural | Dois ou mais | Bom | Não |
| 37 | Rural | Nenhum | Mau | Não |
| 38 | Urbano | Um | Suf | Sim |
| 39 | Rural | Um | Suf | Não |
| 40 | Urbano | Nenhum | Suf | Não |
| 41 | Urbano | Nenhum | Suf | Não |
| 42 | Rural | Nenhum | Bom | Sim |
| 43 | Urbano | Nenhum | Mau | Não |
| 44 | Rural | Dois ou mais | Bom | Não |
| 45 | Urbano | Nenhum | Bom | Não |
| 46 | Rural | Nenhum | MB | Sim |
| 47 | Urbano | Nenhum | Bom | Não |
| 48 | Urbano | Um | Bom | Sim |
| 49 | Rural | Nenhum | Suf | Não |
| 50 | Urbano | Dois ou mais | Mau | Não |
| 51 | Rural | Nenhum | Suf | Não |

| Aluno | Local de residência | Nº de chumbos | Notas Português 3º Período | Fequenta Apoio\explicações |
|---|---|---|---|---|
| 52 | Rural | Um | Bom | Não |
| 53 | Urbano | Nenhum | MB | Sim |
| 54 | Rural | Nenhum | Bom | Não |
| 55 | Urbano | Nenhum | MB | Sim |
| 56 | Rural | Um | Suf | Sim |
| 57 | Urbano | Dois ou mais | Suf | Não |
| 58 | Urbano | Nenhum | Suf | Não |
| 59 | Rural | Nenhum | MB | Sim |
| 60 | Rural | Dois ou mais | Mau | Não |
| 61 | Urbano | Nenhum | Mau | Sim |
| 62 | Urbano | Um | Suf | Não |
| 63 | Rural | Nenhum | Suf | Sim |
| 64 | Rural | Um | Suf | Sim |
| 65 | Urbano | Nenhum | Suf | Não |
| 66 | Urbano | Nenhum | MB | Não |
| 67 | Rural | Nenhum | Suf | Não |
| 68 | Urbano | Nenhum | Bom | Não |
| 69 | Urbano | Nenhum | Bom | Sim |
| 70 | Rural | Nenhum | Suf | Sim |
| 71 | Rural | Dois ou mais | MB | Não |
| 72 | Rural | Nenhum | Bom | Sim |
| 73 | Urbano | Um | Suf | Sim |
| 74 | Rural | Nenhum | Suf | Não |
| 75 | Rural | Nenhum | Bom | Sim |
| 76 | Urbano | Nenhum | Mau | Não |
| 77 | Urbano | Nenhum | Mau | Sim |
| 78 | Urbano | Dois ou mais | Suf | Sim |
| 79 | Urbano | Nenhum | MB | Sim |
| 80 | Rural | Um | Bom | Não |
| 81 | Rural | Dois ou mais | Suf | Sim |
| 82 | Rural | Nenhum | Suf | Não |
| 83 | Urbano | Nenhum | Mau | Sim |
| 84 | Rural | Nenhum | Suf | Sim |
| 85 | Rural | Nenhum | Bom | Não |
| 86 | Urbano | Dois ou mais | Suf | Sim |
| 87 | Rural | Um | Suf | Não |
| 88 | Urbano | Um | Bom | Não |
| 89 | Rural | Nenhum | Mau | Sim |
| 90 | Rural | Nenhum | Suf | Sim |
| 91 | Rural | Um | Suf | Sim |
| 92 | Urbano | Nenhum | Bom | Não |
| 93 | Urbano | Dois ou mais | MB | Não |
| 94 | Rural | Nenhum | Suf | Não |
| 95 | Urbano | Nenhum | Suf | Sim |
| 96 | Urbano | Dois ou mais | Suf | Não |
| 97 | Rural | Um | Suf | Não |
| 98 | Rural | Nenhum | Suf | Sim |
| 99 | Rural | Nenhum | Bom | Sim |
| 100 | Urbano | Um | MB | Não |

| Sexo | Idade | Qualidade das relações com colegas | Nº Faltas |
|---|---|---|---|
| Masculino | Id2 | Muito Bom | Md |
| Feminino | Id3 | Muito Bom | Md |
| Masculino | Id3 | Muito Bom | Md |
| Masculino | Id2 | Muito Bom | Bx |
| Feminino | Id3 | Bom | Md |
| Feminino | Id2 | Mau | Bx |
| Masculino | Id1 | Muito Bom | Bx |
| Feminino | Id2 | Muito Bom | Bx |
| Masculino | Id2 | Muito Bom | Al |
| Feminino | Id2 | Muito Bom | Bx |
| Masculino | Id1 | Razoável | Md |
| Masculino | Id1 | Muito Bom | Md |
| Masculino | Id3 | Muito Bom | Al |
| Feminino | Id2 | Mau | Bx |
| Masculino | Id2 | Bom | Md |
| Feminino | Id2 | Razoável | Bx |
| Feminino | Id1 | Muito Bom | Al |
| Masculino | Id1 | Bom | Bx |
| Masculino | Id3 | Razoável | Bx |
| Feminino | Id3 | Muito Bom | Md |
| Masculino | Id3 | Muito Bom | Bx |
| Masculino | Id3 | Razoável | Bx |
| Masculino | Id1 | Razoável | Bx |
| Feminino | Id1 | Muito Bom | Al |
| Masculino | Id1 | Muito Bom | Bx |
| Masculino | Id2 | Razoável | Bx |
| Feminino | Id2 | Muito Bom | Md |
| Feminino | Id1 | Muito Bom | Al |
| Feminino | Id3 | Muito Bom | Bx |
| Masculino | Id2 | Muito Bom | Bx |
| Masculino | Id2 | Razoável | Bx |
| Masculino | Id1 | Muito Bom | Md |
| Feminino | Id2 | Bom | Bx |
| Masculino | Id1 | Bom | Md |
| Masculino | Id1 | Muito Bom | Al |
| Feminino | Id3 | Muito Bom | Md |
| Feminino | Id3 | Bom | Bx |
| Masculino | Id3 | Bom | Bx |
| Feminino | Id2 | Muito Bom | Md |
| Masculino | Id3 | Muito Bom | Md |
| Feminino | Id1 | Razoável | Bx |
| Masculino | Id3 | Muito Bom | Bx |
| Feminino | Id1 | Muito Bom | Bx |
| Feminino | Id1 | Muito Bom | Bx |
| Feminino | Id3 | Bom | Md |
| Masculino | Id2 | Bom | Bx |
| Masculino | Id1 | Muito Bom | Bx |
| Feminino | Id2 | Muito Bom | Bx |
| Feminino | Id1 | Razoável | Bx |
| Feminino | Id1 | Bom | Bx |
| Feminino | Id2 | Bom | Bx |

| Sexo | Idade | Qualidade das relações com colegas | Nº Faltas |
|---|---|---|---|
| Masculino | Id1 | Razoável | Md |
| Feminino | Id1 | Muito Bom | Md |
| Masculino | Id1 | Muito Bom | Al |
| Feminino | Id1 | Muito Bom | Bx |
| Masculino | Id1 | Muito Bom | Bx |
| Masculino | Id2 | Muito Bom | Bx |
| Masculino | Id3 | Muito Bom | Al |
| Masculino | Id1 | Muito Bom | Bx |
| Masculino | Id1 | Muito Bom | Al |