**Faculdade Ciências Universidade do Porto**

**Árvores de decisão**

**Inteligência Artificial**

**João Reis 202005824**

**Tomás Fontes 202107382**

**Ekaterina Aksenova 202007202**

Maio 2023

# Índice

[Índice 1](#_Toc104497765)

[Introdução 2](#_Toc104497766)

[Árvores de decisão 2](#_Toc104497767)

[Desenvolvimento 3](#_Toc104497768)

[Métricas 3](#_Toc104497769)

[Algoritmos para indução de árvore de decisão 4](#_Toc104497770)

[CART 4](#_Toc104497771)

[CHAID 4](#_Toc104497772)

[ID3 5](#_Toc104497773)

[Implementação 5](#_Toc104497774)

[Resultados 7](#_Toc104497775)

[Conclusão 8](#_Toc104497776)

# Introdução

## Árvores de decisão

Uma árvore de decisão é uma representação de uma função que mapeia um vetor de valores de atributos para um único valor de saída conhecida como “decisão”. Este tipo de árvore chega à sua decisão executando uma sequência de testes, começando na raiz e seguindo o ramo apropriado até que uma folha seja alcançada.

Cada nó interno na árvore corresponde a um valor teste de uma das entradas atributos, os ramos do nó são “rotulados” com os possíveis valores do atributo, e os nós folha especificam qual valor deve ser retornado pela função.

Pode haver diferentes tipos de árvores de decisão, baseando no tipo de variável que temos, podendo ser:

* Árvore de decisão de variável categórica: um atributo categórico assume valores dentro de um conjunto finito, onde não há nenhuma ordem natural. Por exemplo: cor.
* Árvore de decisão de variável contínua: um atributo numérico ou contínuo assume valores dentro dos números reais. Por exemplo: idade, peso, altura. Sendo que para os interesses da nossa aplicação decidimos discretizar os atributos numéricos em intervalos entre valores.

As árvores de decisão são utilizadas em diversas ocasiões, por exemplo:

* Mecanismos de recomendação do cliente - Os clientes que compram determinados produtos podem estar inclinados a comprar algo com uma categoria semelhante ao que estão procurando. Com este mecanismo, por exemplo, os clientes podem ser recomendar outro filme de uma determinada categoria depois de terem terminado de assistir um do mesmo tipo. Os mecanismos de recomendação podem ser desenvolvidos utilizando árvores de decisão, que toma as decisões a partir do gosto dos consumidores ao longo do tempo e criando nós com base nessas decisões.
* Identificando Fatores de Risco para Depressão - Um estudo realizado em 2009 na Austrália acompanhou mais de 6.000 pessoas se elas tinham ou não um transtorno depressivo maior durante um período de quatro anos. Os pesquisadores usaram dados como uso de tabaco, uso de álcool e muito mais para criar uma árvore de decisão que iram ser usados para prever o risco de um transtorno depressivo maior. Decisões e diagnósticos médicos dependem de múltiplas entradas para entender um paciente sobre qual poderá ser a melhor maneira de tratar o mesmo. As árvores de decisão podem ser uma ferramenta valiosa para os profissionais de saúde avaliar os seus pacientes.

# Desenvolvimento

## Métricas

Cada algoritmo que iremos ver posteriormente poderá utilizar um diferente tipo de métrica para selecionar os atributos a colocar na árvore durante a sua construção, estas métricas podem ser:

* Entropia - A entropia é uma medida da aleatoriedade da informação que está a ser processada. Quanto maior for a entropia, mais difícil será tirar quaisquer conclusões a partir dessa informação. O algoritmo ID3 utiliza a entropia para calcular a homogeneidade de uma amostra. Se a amostra for completamente homogénea, a entropia é zero e se a amostra for uma amostra dividida igualmente, tem entropia de um.
* Information gain - A árvore de decisão não faz perguntas aleatórias em cada nó para chegar à conclusão. Fazem as perguntas certas no momento certo para certificarmos de que a entropia é reduzida tanto quanto possível a esse nível em particular. Isto é feito para atingir a homogeneidade em todos os nós de folhas o mais rapidamente possível e vem com uma melhor árvore de classificação que também é simples. Para simplificar, o nosso objetivo é utilizar a entropia para criar uma árvore pequena e simples com uma boa precisão de classificação. Isto é feito através da redução da entropia global na próxima etapa imediata. A fim de reduzir essa entropia, precisamos de compreender qual é o melhor atributo a utilizar para o interrogatório num determinado nó. Isto é feito utilizando uma medida chamada Information Gain.
* Gini index – É calculado subtraindo a soma das probabilidades quadráticas de cada classe. Favorece sobretudo as partições maiores e são muito simples de implementar. Em termos simples, calcula a probabilidade de uma determinada característica selecionada aleatoriamente que foi classificada incorretamente.

O Índice de Gini varia entre 0 e 1, onde 0 representa a pureza da classificação e 1 a distribuição aleatória de elementos entre várias classes. Um Índice de Gini de 0,5 mostra que existe uma distribuição igual dos elementos entre algumas classes.

* Gain Ratio - Information gain é tendencioso para a escolha de atributos com um grande número de valores como nós de raiz. Isto significa que prefere o atributo com um grande número de valores distintos.

Gain ratio que é uma modificação do information gain que reduz o seu enviesamento e é geralmente a melhor opção. O Gain ratio supera o problema do information gain, tendo em conta o número de ramos que resultariam antes de fazer a divisão. Corrige o ganho de informação, tendo em conta a informação intrínseca de uma divisão.

* Chi-Square - O acrónimo CHAID significa "Chi-squared Automatic Interaction Detector". É um dos métodos mais antigos de classificação de árvores. Descobre o significado estatístico entre as diferenças entre os sub-nós e o nó pai. Medimos pela soma dos quadrados de diferenças entre as frequências observadas e esperadas da variável alvo.

Funciona com a variável alvo categórica "Sucesso" ou "Falha". Pode efetuar duas ou mais partições. Mais alto o valor do Qui-quadrado mais alto o significado estatístico das diferenças entre o sub-nó e o nó Pai.

## Algoritmos para indução de árvore de decisão

### CART

Modelo CART, é um algoritmo de árvore de decisão para construir modelos, sendo que os valores alvo têm uma natureza discreta e são conhecidos como modelos de classificação.

Um valor discreto é um conjunto de valores finitos, por exemplo, idade, tamanho, etc. Os modelos em que os valores-alvo são representados por valores contínuos são normalmente chamados de Modelos de Regressão. As variáveis contínuas são variáveis de vírgula flutuante. Estes dois modelos em conjunto chamam-se CART.

CART utiliza o Índice de Gini como matriz de classificação.

### CHAID

CHAID é uma ferramenta utilizada para descobrir a relação entre as variáveis. A análise CHAID constrói uma árvore para ajudar a entender como as variáveis melhor se fundem para explicar o resultado na variável dependente dada.

Na análise CHAID, podem ser utilizados dados nominais, ordinais e contínuos, onde os dados contínuos são divididos em categorias com um número aproximadamente igual de observações.

CHAID cria todas as tabulações cruzadas possíveis para cada dado categórico até que o melhor resultado seja alcançado e nenhuma outra divisão possa ser executada. O desenvolvimento da árvore de decisão começa com a identificação da variável alvo, que seria considerada a raiz. Esta análise divide o alvo em duas ou mais categorias que são chamadas de nós iniciais e depois os nós são divididos utilizando algoritmos estatísticos em nós filhos.

### ID3

A ideia do algoritmo ID3 é construir a árvore de decisão, utilizando uma pesquisa greedy de cima para baixo através dos conjuntos dados para testar cada atributo em cada nó de árvore. A fim de selecionar o atributo mais útil para classificar um dado conjunto, introduzimos um ganho de informação métrica.

Para encontrar uma forma ótima de classificar um conjunto de aprendizagem, o que precisamos de fazer é minimizar as questões colocadas (ou seja, minimizar a profundidade da árvore). Assim, precisamos de alguma função que possa medir quais as questões que proporcionam a divisão mais equilibrada. A informação ganha métrica é uma dessas funções.

O ID3 tem imensas limitações, por exemplo:

* Por causa de utilizar um algoritmo greedy, não irá garantir uma solução óptima.
* Este algoritmo produz normalmente árvores pequenas, mas nem sempre produz a árvore mais pequena possível.
* O ID3 é mais difícil de utilizar em dados contínuos (se os valores de qualquer atributo forem contínuos, então há muito mais lugares para dividir os dados sobre este atributo, e procurar o melhor valor para dividir pode ser demorado).

## Implementação

Para este trabalho escolhemos C++ como a linguagem de implementação do projeto, dado que todos os elementos do grupo já tinham alguma familiaridade com o seu funcionamento. Esta linguagem, sendo orientada a objetos, também proporciona uma grande facilidade em criar as estruturas de dados que se mostraram fundamentais neste projeto.

A nossa implementação deste algoritmo está estruturada em 3 partes distintas: a leitura e armazenamento dos dados, a construção da árvore de decisão, e a leitura e classificação dos testes.

Para a leitura dos dados implementámos algumas classes para tornar a utilização dos dados o mais fácil possível. Criámos as classes Val, Example e Atribute. Val contem uma string com o nome real do valor e um contador de ocorrências nos exemplos de aprendizagem. Example contem o ID do exemplo, um vetor que associa índices de atributos ao seu respetivo valor no exemplo e a classe associada. Finalmente, Atribute contem o nome do atributo, um vetor de Vals para os valores possíveis do mesmo, e um vetor value\_limits que é só utilizado para atributos numéricos.

Os atributos são guardados num vetor global da classe Atribute, enquanto que os testes e os exemplos são mantidos em vetores globais da classe Example. Na nossa implementação, os testes são tratados como exemplos sem classe até esta ser atribuida.

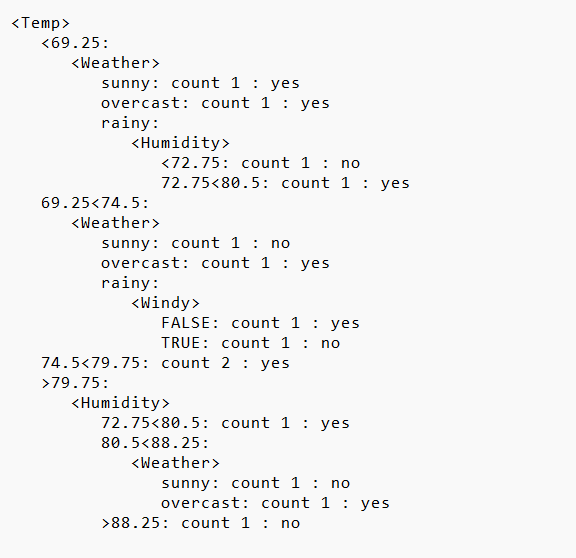
Para a construção de uma árvore nós implementámos mais uma classe Node que contem um índice para o atributo respetivo e outro para o valor respetivo, assim como um caracter para indicar o tipo de node que é (a se for um atributo, v se for um valor e l se for uma folha). Nodes têm também dois vetores, um de índices de atributos que ainda não foram considerados, e outro de índices de exemplos que respeitam as restrições desse nó e dos seus predecessores. Finalmente, os Nodes também guardam uma classe, que só é relevante para os nós folha, assim como um apontador para o seu nó pai e um vetor de apontadores para os seus filhos.

Os nós são expandidos com base no seu tipo. Nodes do tipo atributo são expandidos criando um nó filho do tipo v para cada valor possível e restringindo os exemplos desses nós para aqueles que respeitam esse valor, se não houver nenhum exemplo que sirva o nó é apagado. Os nós do tipo valor são expandidos para um nó folha se e só se todos os seus exemplos tiverem a mesma classe, caso contrário o seu nó filho será do tipo a cujo atributo é escolhido por valor máximo de entropia a partir do vetor de atributos ainda não explorados.

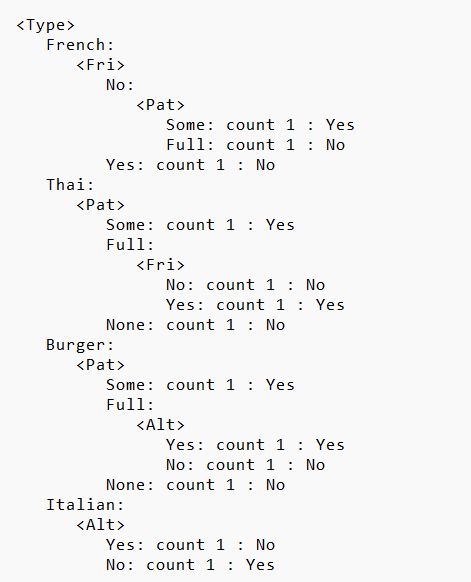
Os testes são avaliados explorando a árvore construída comparando apenas os valores do nós v aos valores do teste e, chegando a um nó folha, é atribuída ao teste a sua classe.

## Resultados

Obtemos estes resultados para o weather:



E por fim, obtivemos estes resultados para o restaurante:



# Conclusão

Em suma, as árvores de decisão são úteis para categorizar resultados dos quais os atributos podem ser classificados em relação a critérios conhecidos para determinar uma categoria final.