

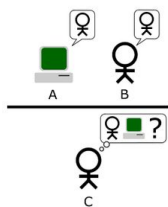
P1 - Inteligência artificial

Este documento apresenta um resumo dos principais tópicos da matéria de inteligência artificial.

1. INTRODUÇÃO

De acordo com RUSSEL e NORVIN (2010), a definição de inteligência artificial não é direta, e esta pode ser dividida em quatro categorias principais, são elas:

- Sistemas que pensam como humanos: Sistemas que realizam a automatização de atividades que nós, como seres humanos, associamos com o pensamento humano em uma tomada de decisões (BELLMAN, 1978);
 - Veja que, nestes sistemas é preciso entender como os seres humanos pensam.
- Sistemas que agem como humanos: O estudo de como fazer computadores realizarem tarefas que, no momento, pessoas são melhores (RICH, 1991)
 - O teste de turing entra aqui, com o objetivo de medir a capacidade da máquina de se comportar como um ser humano. Neste teste existe uma máquina (A), uma pessoa (B) e um juiz (C), o juiz busca identificar o que é máquina ou homem vide Figura TT;



- O captcha é um teste de turing.
- Sistemas que pensam racionalmente: O estudo das computações que fazem possível perceber, pensar e agir (WINSTON, 1992);
- Sistemas que agem racionalmente: Inteligência computacional é o estudo do projeto de agentes inteligentes (POOLE, 1998).

Com isto, pode-se entender também quais são os objetivos gerais da inteligência artificial, estes são listados abaixo:

- **Compreender** o comportamento inteligente;
- **Reproduzir** o comportamento inteligente;
- **Construir** sistemas inteligentes para realizar tarefas, mesmo estas sendo melhores realizadas por seres humanos ou que não possuam solução satisfatória através da computação convencional.

Além das quatro categorias principais, apresentadas anteriormente, é possível identificar também que, dentro da área de inteligência artificial existem paradigmas para a criação e aplicação de algoritmos, dentre estes, cita-se como principais:

- Paradigma simbólico: Este é o paradigma que dá ênfase a sistemas baseados em conhecimento (Sistemas especialistas). Este paradigma é fundamentado em metáforas linguísticas;
- Paradigma estatístico/probabilístico: Com ênfase em *Machine Learning*, estuda os principais algoritmos baseados na abordagem estatística/probabilística. Métodos estatísticos são empregados aqui, como as redes bayesianas, sistemas difusos e outros;
- Paradigma conexionista: Com ênfase em *Deep Learning*, estudo da estrutura do neurônio artificial, estudo e construção de redes neurais artificiais. Utiliza metáforas cerebrais. Tem como premissa a suposição de que o comportamento inteligente está relacionado com a dinâmica das conexões entre pequenos nós denominados neurônios, onde tal dinâmica é capaz de representar o conhecimento;
- Paradigma Evolucionista: Tem como base conceitual os algoritmos genéticos. Trabalha com metáforas da natureza.

Comportamento inteligente.

Comportamento inteligente dispositivos em ambientes complexos:

- percepção
- raciocínio

- aprendizado
- comunicação
- ação e planejamento

Sistema inteligente:

Um sistema inteligente deve ser capaz de:

- adaptar-se a novas situações
- raciocinar
- entender relações entre fatos
- descobrir significados
- aprender com base na experiência

2. SISTEMAS INTELIGENTES

Sistemas inteligentes são aqueles com capacidade de exibição de comportamento inteligente, nestes ações inteligentes são executadas sobre o domínio específico de algum problema.

Dentre as principais características dos sistemas inteligentes, nota-se como principais:

- Conhecimento específico do domínio do problema: Este é um requisito indispensável para a solução de problemas complexos;
- Habilidade na utilização do conhecimento: Permite a execução de tarefas complexas (Incluindo as ações inteligentes), fazendo até mesmo a intervenção humana não ser necessária.

No contexto de sistemas inteligentes, a inteligência artificial busca capacitar estes sistemas para a realização de ações inteligentes.

2.1. Ações inteligentes



Ação inteligente: Uma ação inteligente deve incorporar dois elementos indispensáveis: o **conhecimento** e o **raciocínio** que são características do comportamento do sistema mais perfeito, o próprio ser humano.

Conhecimento: Conhecimento em sistemas inteligentes, baseia-se na combinação de informações para que possa ser utilizada de várias formas úteis, este procedimento de combinação de informações, implica na obtenção de resultados e conclusões para a tomada de decisões estratégicas.

Raciocínio: Baseia-se na maneira de como sistemas inteligentes chegam as conclusões interpretando o conhecimento adquirido ou gerando um novo conhecimento.

Aquisição de conhecimento: Todo sistema inteligente deve ter a capacidade de adquirir conhecimento ou capacidade de aprendizagem, é um atributo fundamental e indispensável.

A aquisição de conhecimento se dá por meio de combinações entre informações

Estratégia de Raciocínio: Também chamado de **estratégia de inferência** corresponde a dois métodos chamados **inferência indutiva** e **inferência dedutiva**, que são utilizados para obter conclusões da parte do sistema inteligente.

2.2. Tipos de sistemas inteligentes

Existem diferentes tipos de sistemas inteligentes, muitos deles contendo especializações, para a resolução de diferentes problemas, a Figura abaixo apresenta uma hierarquia de sistemas até os sistemas especialistas.



2.2.1. Sistemas baseados em conhecimento

São programas baseados em informações e conhecimento humano, estes que são tratados de forma inteligente pelo sistema a fim de se chegar a uma resposta que necessita de alto conhecimento humano.

Um sistema baseado em conhecimento possui

- Representação de conhecimento: Meios disponíveis para armazenar e processar o conhecimento existente;
- Inferência: Novas conclusões que o sistema pode ter a partir de seu conhecimento;
- Aquisição de conhecimento: Etapa onde o conhecimento adquirido é adicionado ao sistema;
- Conhecimento explícito: Motivo da conclusão é explicado ao usuário pelo sistema.

Vantagens:

- Menor custo para o usuário;
- Acessível a qualquer hora;
- Normalmente mais rápido que os especialistas humanos;
- Confiabilidade no sistema pode ser maior que a os especialistas humanos
- Não há distrações, fadiga, envolvimento emocional;
- Explicação dos passos do raciocínio que conduziram a uma determinada conclusão, estão
- explicados claramente.

Desvantagens:

- Conhecimento limitado e Superficial;
- Não há uma compreensão profunda dos conceitos envolvidos e suas relações
- mútuas;

- Não possuem senso comum;
- Conhece somente o que lhe foi explicitamente ensinado, não tem conhecimento do que não conhece;
- Pode não ser encontrado o método mais apropriado para um problema em particular;
- Alguns problemas “fáceis” demandam muito processamento de máquina;
- Usuário não tem 100 % de confiança no sistema.

Base de conhecimento: Funcionalidades do sistema especialista que representam seu conhecimento, com base em regras e fatos.

Máquina de inferência: É responsável pelo desenvolvimento do raciocínio baseado nas informações obtidas e no conhecimento representado na base de conhecimentos. Existem algumas linhas de seguimento possíveis em sistemas baseados em conhecimento, estas são:

Backward chaining: Este processo parte da suposição de que cada provável solução é verdadeira. Feito isso, tenta-se reunir evidências que comprovem ser correta a solução previamente considerada. É equivalente a pergunta: “É possível chegar em alguma conclusão com os dados informados?”

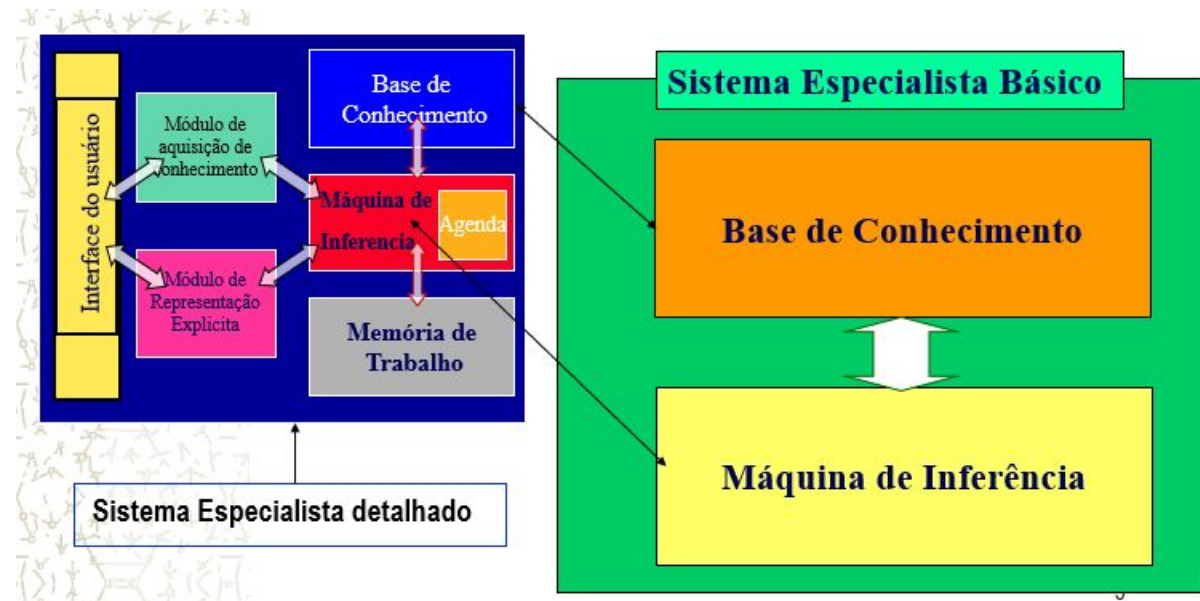
Forward chaining: Neste processo chega-se a uma solução para o problema a partir das informações fornecidas pelo usuário. Este processo consiste em analisar esses dados, baseando-se no conhecimento, até chegar a uma conclusão. É equivalente a pergunta: "Esses dados chegam a alguma conclusão?"

2.2.1.1 Sistemas baseados em conhecimento X sistemas convencionais

Sistemas Convencionais	Sistemas Baseados em Conhecimento
Estrutura de Dados	Heurística : É o conhecimento de regras práticas que auxiliam a reduzir o esforço de busca por soluções
Dados e Relações entre Dados	
Tipicamente usa algoritmos determinísticos	
Conhecimento embutido no código do programa	
Explicação do raciocínio é difícil	

2.2. Os sistemas especialistas

Sistemas especialistas, são sistemas que tem como objetivo simular o raciocínio de um profissional em alguma área de conhecimento específica. A arquitetura mais comum de um sistema especialista é a que envolve regras de produção na sua base de conhecimento.



Estrutura básica de um SBC

Base de conhecimento: Contém **informações essenciais** sobre o domínio problema, normalmente representada por fatos de regra de produção.

Máquina de inferência: A máquina de inferência consiste no procedimento que infere um novo conhecimento, a partir da base de conhecimento (regras) e da informação fornecida pelo usuário

Estrutura das Regras de produção

As regras de produção de um sistema especialista, consta em duas partes chamadas de **antecedente** e **consequente**:

Antecedente: As regras de produção denominadas como antecedente, são formadas por uma série de condições prévias, onde essas condições estão unidas por conectivos lógicos dos tipo (E) e (OU).

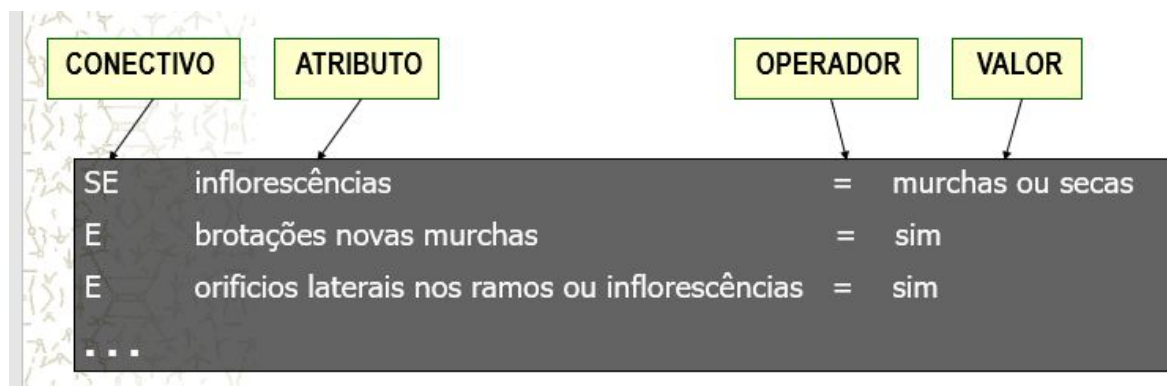
Consequente: As regras de produção denominadas de consequente (também chamadas de conclusão), são correspondentes aos resultados das regras decorrentes do cumprimento das condições prévias. A conclusão das condições prévias são determinadas em um grau de confiança em %

Cabeça de regra: As regras de produção consequente são denominadas **cabeça de regra**

Caudas da regra: As regras de produção antecedente são denominadas **cauda de regra**.

Sintaxe de uma regra de produção

A estrutura da cauda (**antecedente**) deve obedecer o seguinte modelo:



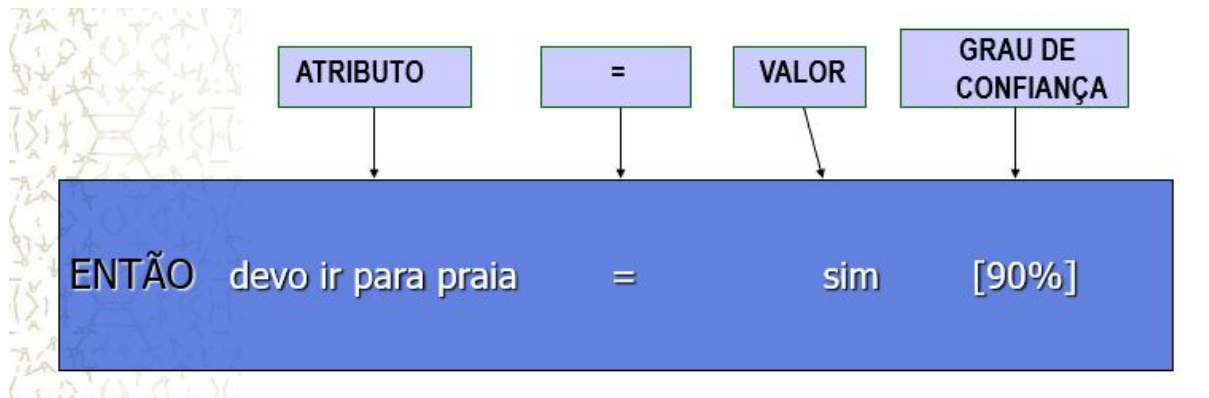
Conectivo: É um dos seguintes elementos da lógica clássica: **NÃO E OU**. Esses conectivos use a **sentença** ao conjunto de **premissas** que formam a cauda (**antecedente**) de uma regra

Atributo: Atributos, são variáveis capazes de assumir uma ou múltiplas instanciações no decorrer da consulta à base de conhecimentos.

Operador: É uma ligação entre o atributo e o valor da premissa que define o tipo de comparação a ser realizada. São operadores relacionais: =, >, <, <>, entre outros.

Valor: É um item de uma lista a qual foi previamente criada e relacionada a um atributo

A estrutura da cabeça (**conclusão**) deve obedecer o seguinte modelo:



Atributo: Corresponde as mesmas características do atributo das caudas

= Operador de atribuição: É um operador de atribuição e não de igualdade. Ou seja a variável designada pelo atributo na cabeça de regra é sempre instanciada a um valor.

valor: Corresponde ao mesmo valor utilizado nos antecedentes.

Grau de confiança: É um valor de porcentagem que mostra qual é a confiabilidade daquela conclusão de regra. As porcentagens variam de 0% a 100%

Códigos de exemplo utilizando PyKnow.

```
from random import choice
from pyknow import *

class Light(Fact):
    """Info about the traffic light."""
    pass

class RobotCrossStreet(KnowledgeEngine):
    @Rule(Light(color='green'))
    def green_light(self):
        print("Walk")

    @Rule(Light(color='red'))
    def red_light(self):
        print("Don't walk")
    @Rule(AS.light << Light(color=L('yellow') | L('blinking-yellow'))))
    def cautious(self, light):
        print("Be cautious because light is", light["color"])

engine = RobotCrossStreet()
engine.reset()
```

```

from random import choice
from pyknow import *

class AmanhaPodeChover(Fact):
    pass
class TenhoDinheiro(Fact):
    pass
class TenhoTempo(Fact):
    pass
class DevoIrAPraia(Fact):
    pass
class NaoVouSair(Fact):
    pass
class NenhumaEmergencia(Fact):
    pass

class DevoIrAPraiaMachine(KnowledgeEngine):
    @Rule(NOT(NaoVouSair(fato=W()))):
    def ask_data(self):
        action = input("Vai sair? (s/n)")
        action = (False if action == 's' else True)
        print(action)
        self.declare(NaoVouSair(fato=action))

    @Rule(NOT(NenhumaEmergencia(fato=W()))):
    def ask_data2(self):
        action = input("Tem emergencia? (s/n)")
        action = (False if action == 's' else True)
        print(action)
        self.declare(NenhumaEmergencia(fato=action))

    @Rule(NOT(AmanhaPodeChover(fato=W()))):
    def ask_data3(self):
        action = input("Amanha pode chover? (s/n)")
        action = (True if action == 's' else False)
        print(action)
        self.declare(AmanhaPodeChover(fato=action))

    @Rule(NOT(TenhoTempo(fato=W()))):
    def ask_data4(self):
        action = input("Tenho tempo? (s/n)")
        action = (True if action == 's' else False)
        print(action)
        self.declare(TenhoTempo(fato=action))

    @Rule(AND(
        AmanhaPodeChover(fato=False),
        TenhoDinheiro(fato=True),
        TenhoTempo(fato=True),
    ))
    def devoIr(self):
        print("Devo ir a praia")

    @Rule(AND(
        NaoVouSair(fato=True),
        NenhumaEmergencia(fato=True)
    ))
    def tenhoDinheiro(self):
        self.declare(TenhoDinheiro(fato=True))

engine = DevoIrAPraiaMachine()
engine.reset()
engine.run()

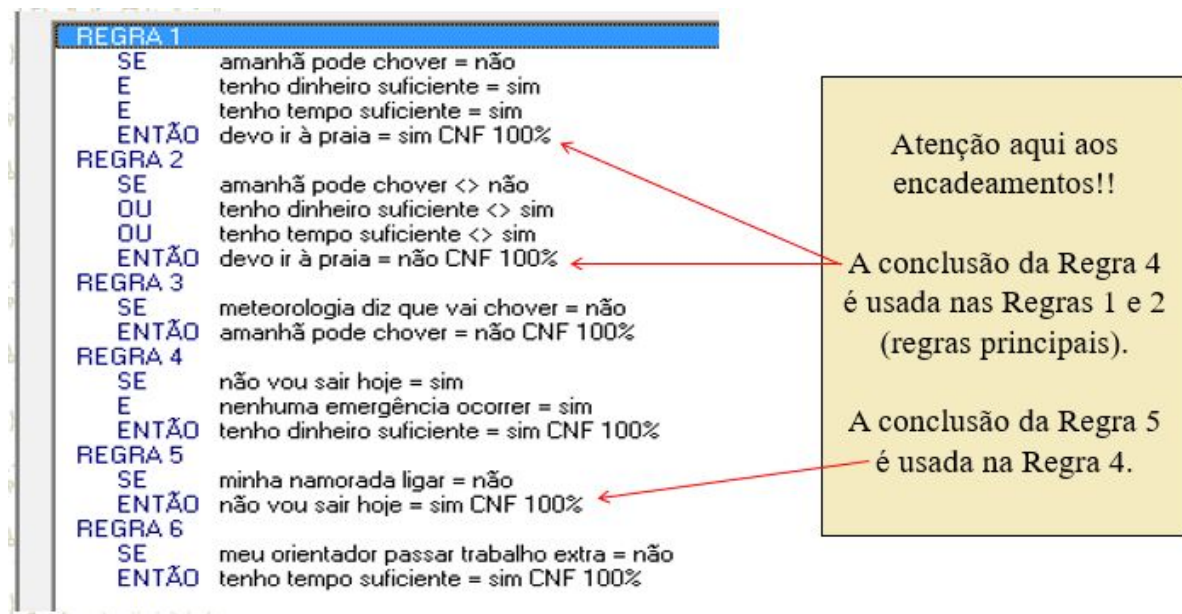
```

Implementação do DevoIrAPraia em PyKnow

Exemplo de backward chaining

Tomando em base escolhas e fatores de regras passadas, conclui-se um resultado, no qual o mesmo é utilizado na próxima regra (não sendo delimitado como resultado final, mas sim apenas uma continuação para uma tomada de decisão mais específica)

Exemplo:



3. ÁRVORE DE DECISÃO

Nem sempre será possível articular o conhecimento que é necessário para a construção de um sistema especialista. Em certos casos, este conhecimento simplesmente não existe, desta forma faz-se a utilização de algoritmos que aprendem este conhecimento necessário. Estes algoritmos são conhecimentos normalmente por algoritmos de *Machine Learning* (ML).

3.1. Aprendizado

O ato de aprender, citado anteriormente refere-se ao ato de memorizar alguma coisa, aprender fatos por meio de observação e exploração. É dito que um sistema aprende quando, ele fica melhor levando em consideração situações anteriores.

A generalização do conhecimento, para os algoritmos de *Machine Learning* só aparecem a partir de experiências. Pode-se entender que, o aprendizado dentro do *Machine Learning* é completamente empírico. O grande problema nesta etapa de aprender de forma empírica é justamente com os dados, para ter o que foi citado como “experiências anteriores” o algoritmo deve utilizar muitos dados, o que nem sempre está disponível no mundo real, por isto considera-se que, algoritmos de ML são heurísticos, por sempre estarem mudando seus objetivos e estratégias de acordo com o conjunto de dados disponível.

Dentre os problemas de aprendizado pode-se encontrar: Reconhecimento de assinaturas e classificação de mensagens.

3.2. Tarefas

Em *Machine Learning* trabalha-se sempre com tarefas, estas que podem ser interpretadas como algum tipo de relacionamento entre os dados, que serão estabelecidos para obter conhecimento passível de interpretação humana. Alguns tipos de tarefas:

- Regressão;
- Agrupamento;

- Classificação.

3.3. A árvore de decisão

Dentre os diversos algoritmos de ML, encontra-se a árvore de decisão, esta que é uma metodologia que permite extrair conhecimento presente nos dados, criada recursivamente levando em consideração algum índice de verificação. Esta é uma metodologia de computação simbólica, já que as árvores aprendem situações descritas através de sentenças lógicas.

As árvores de decisão podem ser definidas como: “Uma árvore de decisão, representa a disjunção de conjunções das restrições nos valores dos atributos que descrevem a situação que está sendo aprendida”. Para que esta afirmação seja entendida, vejamos algumas características das árvores de decisão.

Primeiro, é importante entender que, cada sentença lógica criada pela árvore de decisão representa uma conjunção (E), e que, estas conjunções são disjuntas (OU) entre si, ou seja, ao definir as regras, pode apenas seguir um caminho, ou outro (Uma regra ou outra).

3.3.1. Composição de uma árvore de decisão

Para lembrar dos componentes que compõem uma árvore de decisão, tente se lembrar de uma árvore binária, a estrutura é exatamente a mesma, possuindo nós de decisão (Contendo regras de separação dos dados) e nós folha (Nós fim)

3.3.2. Um exemplo prático da criação de uma árvore de decisão

Árvores de decisão são métodos que focam no aprendizado de máquinas, cujo quais são supervisionados, utilizados principalmente em tarefas de classificação.

Pense comigo, você precisa levar uma bebida para uma festa, mentalmente você caracteriza as pessoas que estão nessa festa, se a maioria é adulto, se existem muitas crianças, se os adultos presentes nesta festa bebem algum tipo de bebida alcoólica, enfim, são inúmeras as possibilidades de classificar a melhor bebida para ser levada através de perguntas simples que se derivam de outras perguntas e assim sucessivamente. Viu? Sem querer a partir de vários fatores, foi

possível pensar em uma lógica e tomar uma decisão, mentalmente sem perceber, você cria uma árvore de decisão, possuindo vários nós de decisão e diferentes nós folha.

3.3.3. Problemas resolvidos com árvores de decisão

As árvores de decisão podem ser aplicadas em problemas de classificação, regressão, isto variando com o *ensemble* que está sendo aplicado. Neste documento, o escopo apresentado será apenas o de regressão.

3.3.4. Entropia, Gini e ganho de informação

Com o entendimento conceitual das árvores de decisão, será apresentado agora, os conceitos utilizados para a geração de uma árvore de decisão. Conforme explicado anteriormente, a geração de uma árvore de decisão é feita recursivamente, porém há técnicas que são aplicadas durante este processo de geração.

Para entender estas técnicas, inicialmente será apresentado os conceitos de entropia e ganho de informação.

Basicamente a entropia representa o nível de “bagunça”, ou o nível de confusão do sistema/conjunto. Este é um índice utilizado para calcular o grau de impureza da coleção que está sendo avaliada.

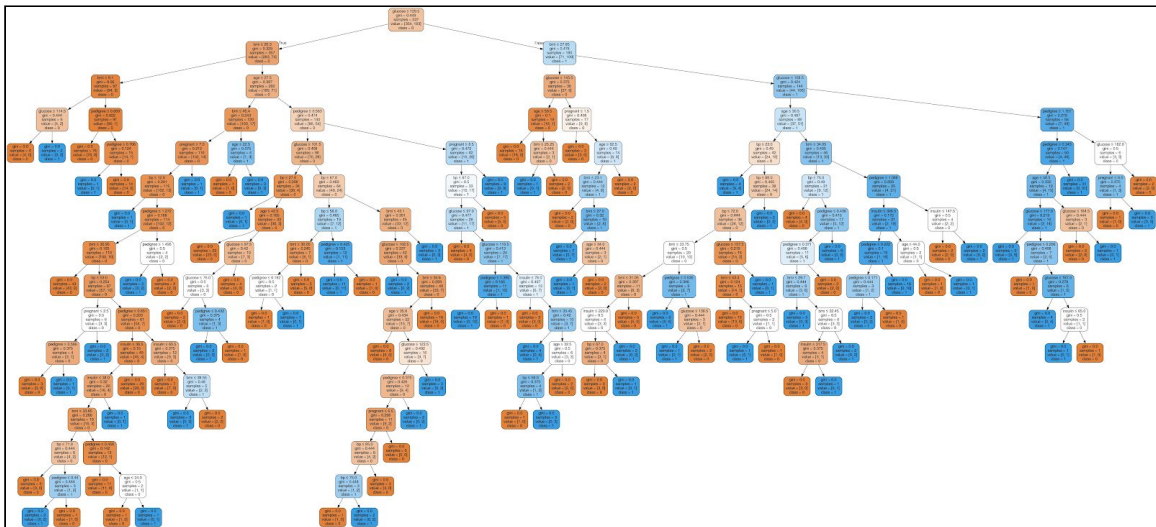
Com a conceituação da entropia, o ganho de informação passa a fazer sentido, já que, este representa a redução da entropia causada pela partição de um conjunto de dados de acordo com um atributo A qualquer. A definição do ganho de informação é apresentada abaixo:

$\text{GanhoDeInformacao} = \text{EntropiaDoConjunto} - \text{EntropiaDoAtributoAQualquer}$

Além da utilização da entropia, que calcula o grau de impureza dos dados pode também ser utilizado o índice gini, que ao contrário da entropia não calcula impureza, mas sim a probabilidade da classificação apresentar problemas.

3.3.5. Poda da árvore

Uma das grandes vantagens das árvores de decisão, é a grande facilidade de interpretação que o método traz, que diferente de outros algoritmos é extremamente visual. Porém, há casos em que, a árvore acaba ficando muito grande e complexa, veja a Figura abaixo.



Isto na maioria das vezes não é bom, os resultados são problemáticos de entender e ainda, podem estar representando um *overfitting* do modelo criado sobre os dados que estão sendo utilizados.

Para resolver este problema aplica-se a técnica de poda (*Prunning*), esta técnica basicamente remove certas partes da árvore, normalmente um conjunto de nós advindos de uma determinada regra, por apenas um nó folha, desta forma limita-se o tamanho da árvore e os problemas apresentados anteriormente são evitados. Os nós selecionados para poda são normalmente aqueles que, têm seu erro maior que o de apenas um nó de representação.

Vale lembrar que esta técnica é utilizada para garantir uma premissa básica das árvores de decisão, a busca pelo menor modelo, que apresente os resultados mais condizentes e concretos.

3.3.6. Matriz de confusão

Anteriormente foi citado que neste documento o foco seria as árvores de decisão utilizadas para a classificação, pois bem, esta subseção apresenta uma forma de avaliação dos resultados gerados pela árvore de decisão, e então, verificar se a classificação feita pela árvore de decisão está correta.

Para isto, utiliza-se comumente a matriz de confusão. Nesta técnica o que é feito basicamente é a criação de uma matriz que relaciona os valores que deveriam ser classificados com os valores que realmente foram classificados. Veja como exemplo a Figura abaixo.

		Prediction	
		Cat	Dog
Actual	Cat	15	35
	Dog	40	10

Veja que, no eixo Y da matriz, existem os valores reais, e no eixo X os valores classificados pelo modelo. Assim pode-se entender que, o modelo classificou 15 gatos quando era para ser gato, e 35 cachorros quando era para ser gato. E o mesmo processo de verificação é verdadeiro para a linha *Dog*.

Regressão linear

O que é:

Regressão linear é o processo de traçar uma reta através dos dados em um diagrama de dispersão.

Sendo assim, pode-se dizer que é o processo de traçar uma reta através dos dados em um diagrama de dispersão, em que a reta se dá pelo correlacionamentos entre as variáveis independente com a variável dependente.

Pra que serve (contexto geral):

Para estimar a condicional (valor esperado) de uma variável Y, dados os valores de algumas outras variáveis X.

A partir de um modelo de regressão linear, conseguimos saber a correlação entre uma variável dependente em relação com as variáveis independentes. O resultante deste modelo são os valores para as variáveis A e B da equação da reta:

$$Y = Ax + B$$

Regressão linear, em cima da correlação das variáveis, indica o:

- Coeficiente (variável A);
- Intercepto (variável B).

Em que Y é a variável dependente, ou seja, é a variável que depende de uma variável independente para ter um valor. Vale ressaltar que existem diversas formas para a criação de um modelo de regressão.

Pra que serve (contexto de Inteligência Artificial):

Regressão linear trata-se de um modelo de regressão utilizado para realizar previsões entre as variáveis de um conjunto de dados (não confundir com modelo de classificação, pois o mesmo não agrupa atributos e comportamento em relação a alguma natureza).

Métodos de seleção (M5, Greedy)

Para a criação de um modelo de regressão linear, é utilizado um método de seleção de atributos. Os mais usuais são:

- **M5:** Percorre os atributos e eliminando os que possuem menor correlação.
- **Greedy:** Método de busca gulosa, realizado utilizando a métrica de informação Akaike (definição extraída da documentação do Weka).

Correlação

Correlação é a medida do grau de associação entre duas variáveis, a fim de saber o quanto um variável possui relação com outra.

Gráfico de Dispersão

O gráfico de dispersão é utilizado normalmente para verificar de forma gráfica, a relação entre duas variáveis.

As relações das variáveis se resume aos tipos apresentados na Figura abaixo.

