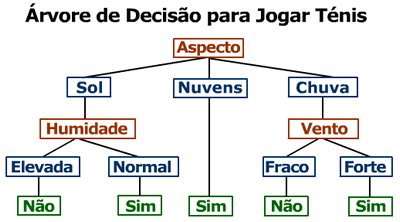
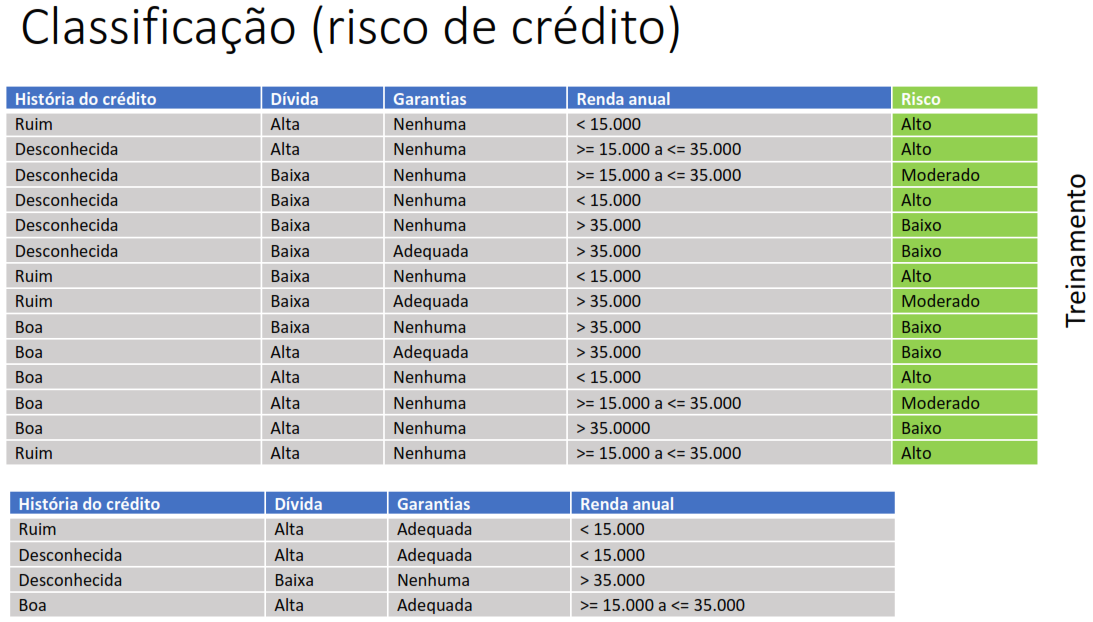
09\_07\_2020\_Arvore\_Decisao

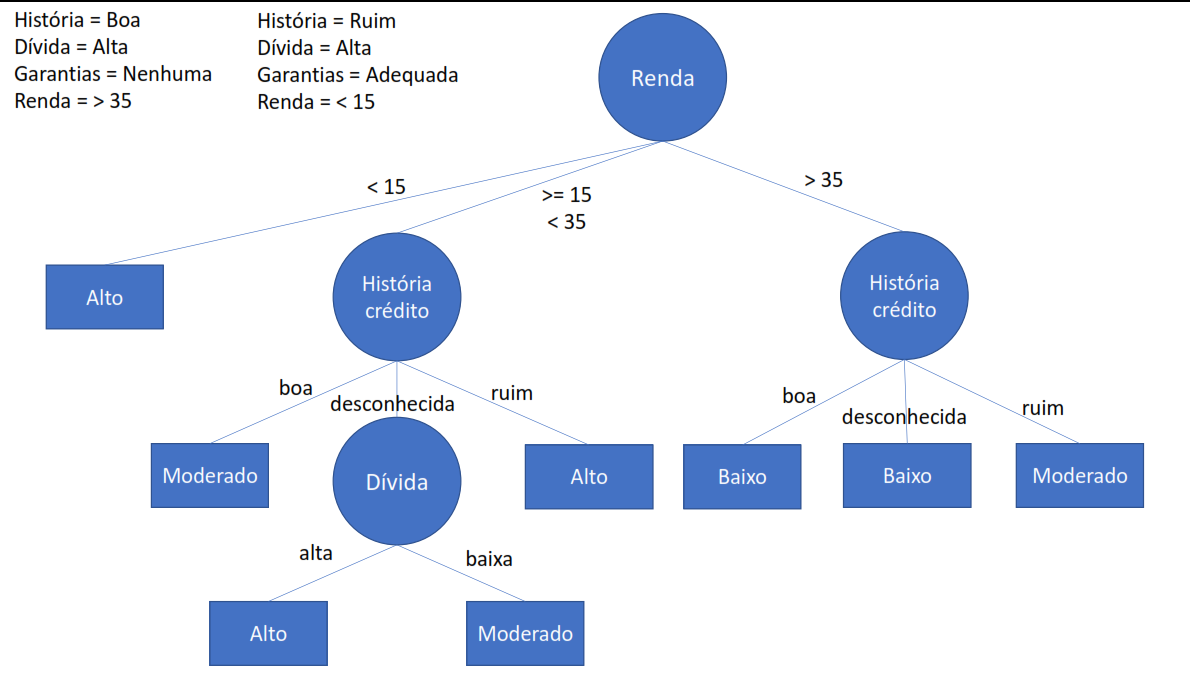


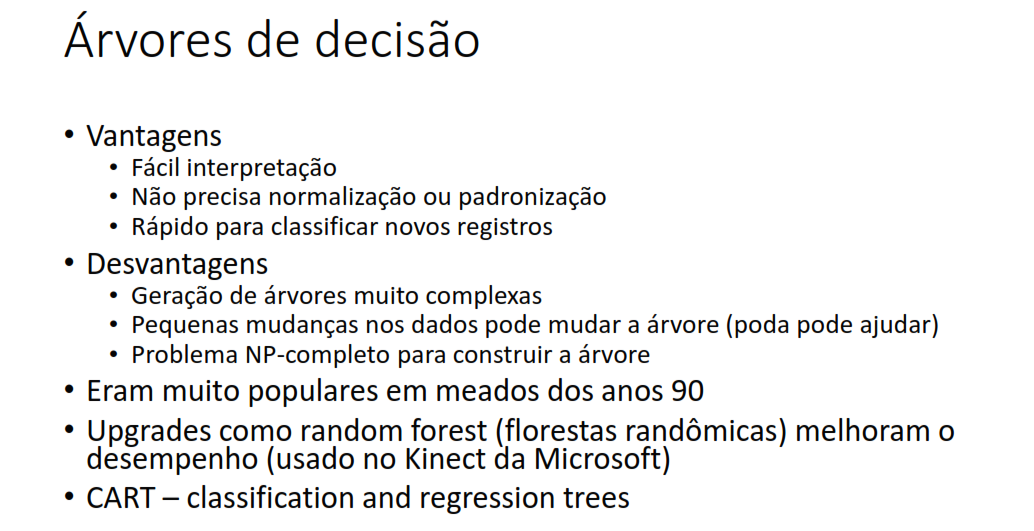
O aprendizado indutivo de árvores de decisão é geralmente dividido em aprendizado **supervisionado** e **não-supervisionado**.

As **Árvores de Decisão** são um dos modelos mais práticos e mais usados em inferência indutiva. Este método representa funções como árvores de decisão. Estas árvores são treinadas de acordo com um conjunto de treino (exemplos previamente classificados) e posteriormente, outros exemplos são classificados de acordo com essa mesma árvore. Para a construção destas árvores são usados algoritmos como o ID3, ASSISTANT e C4.5.

**Árvores de decisão**







**Arvore de Decisão**

* Deve-se considera o uso de árvores de decisão em situações onde:
* As instâncias são descritas por pares atributo-valor;
* Alvo é de valor discreto;
* Os exemplos de treino poderão ter erro;
* Faltam valores nos atributos;
* Exemplos:  
  - Diagnósticos médicos;  
  - Análises de risco de crédito;  
  -Glosa de plano de saúde

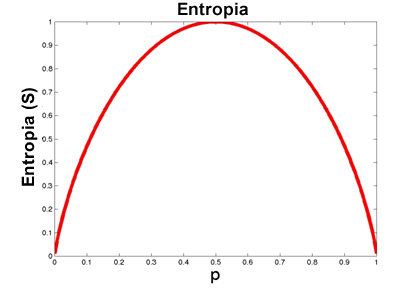
Entropia - A entropia de um conjunto pode ser definida como sendo o grau de pureza desse conjunto.

O ganho (gain) é define a redução na entropia. Ganho(S,A) significa a redução esperada na entropia de S, ordenando pelo atributo A.

Dado um conjunto *S*, com instâncias pertencentes à classe *i*, com probabilidade pi, temos:

http://web.tecnico.ulisboa.pt/ana.freitas/bioinformatics.ath.cx/bioinformatics.ath.cx/uploads/RTEmagicC_arv_dec7.gif.gif    (1)

No exemplo apenas existem duas classes de classificação, ou seja, "Jogar Ténis" (positivo, +) ou "Não Jogar Ténis" (negativo, -). Assim sendo, o valor da entropia varia de acordo com o gráfico:



Onde:

* S é o conjunto de exemplo de treino;
* p+ é a porção de exemplos positivos;
* p- é a porção de exemplos negativos;
* A entropia é dada pelo desdobramento da equação 1  
  http://web.tecnico.ulisboa.pt/ana.freitas/bioinformatics.ath.cx/bioinformatics.ath.cx/uploads/RTEmagicC_arv_dec8.gif.gif

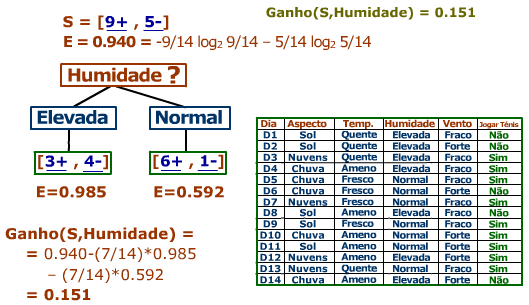
**Algoritmo ID3**

O algoritmo ID3 (inductive decision tree) é dos mais utilizados para a construção de árvores de decisão. Este algoritmo segue os seguintes passos:

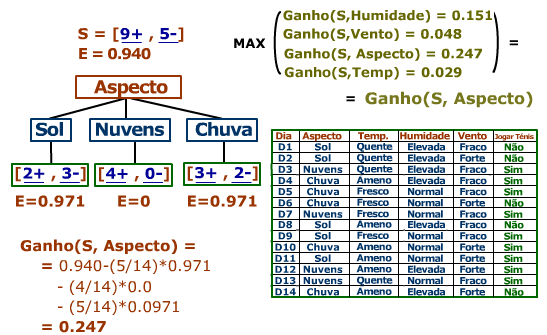
1. Começar com todos os exemplos de treino;
2. Escolher o teste (atributo) que melhor divide os exemplos, ou seja agrupar exemplos da mesma classe ou exemplos semelhantes;
3. Para o atributo escolhido, criar um nó filho para cada valor possível do atributo;
4. Transportar os exemplos para cada filho tendo em conta o valor do filho;
5. Repetir o procedimento para cada filho não "puro". Um filho é puro quando cada atributo X tem o mesmo valor em todos os exemplos.

**Escolha do Melhor Atributo**

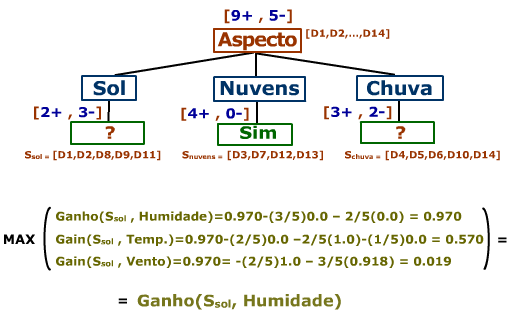
Para responder à pergunta anterior, "Como escolher o melhor atributo" é usado o ganho. Em cada iteração do algoritmo é escolhido o atributo que apresente uma maior ganho.

Para o primeiro passo são analisados todos os atributos, começando pela **Humidade**, por exemplo:

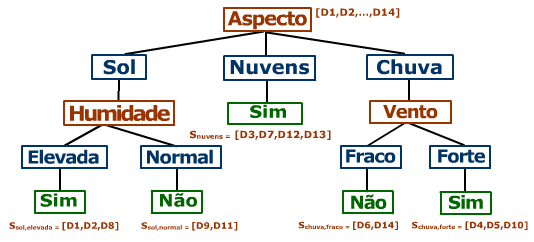
Calculando o ganho para todos os atributos, verificamos que o tem maior ganho é o **Aspecto.**

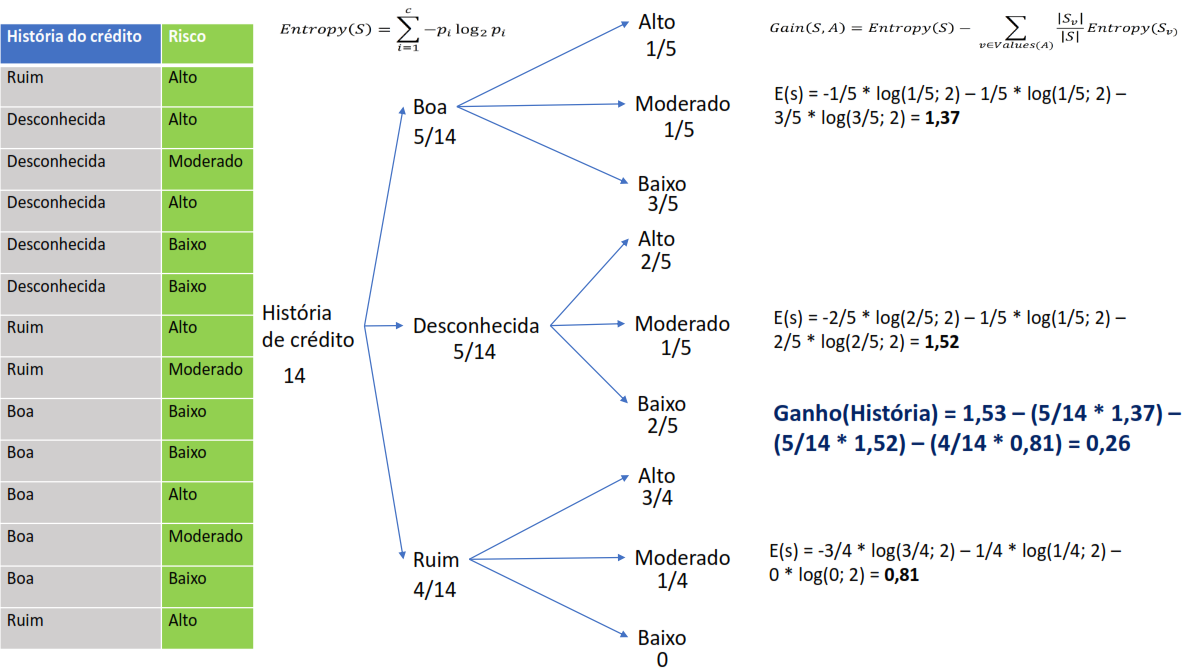


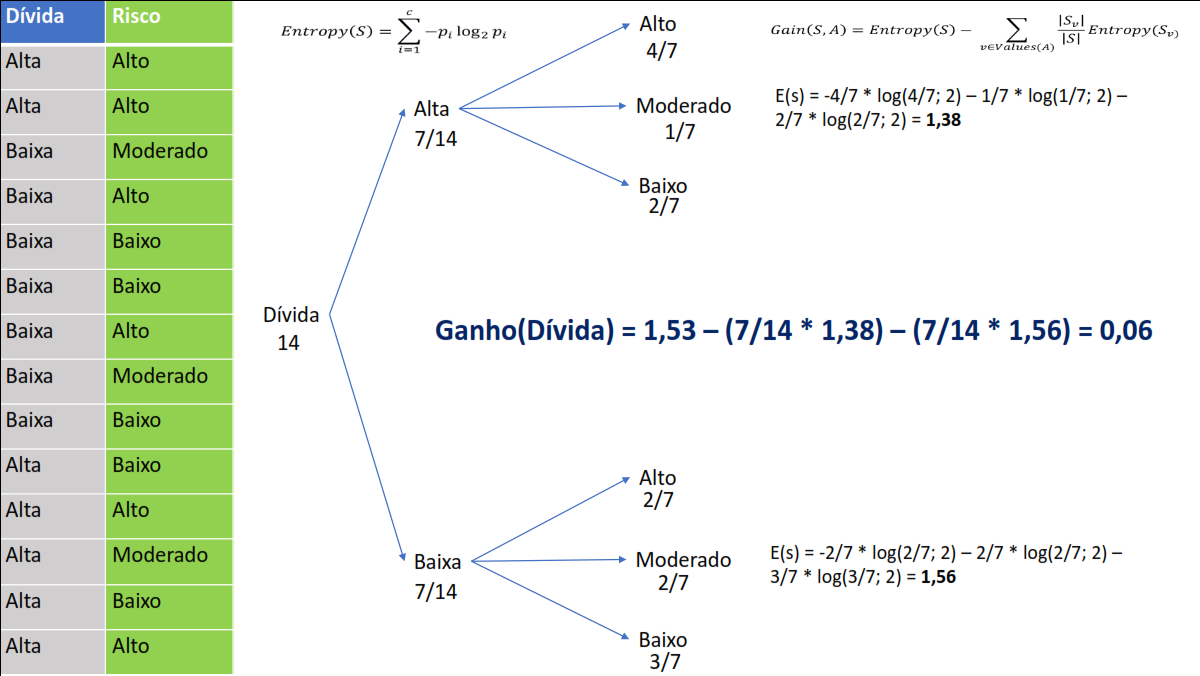
No próximo passo o atributo **Aspecto** já não é tido em conta:

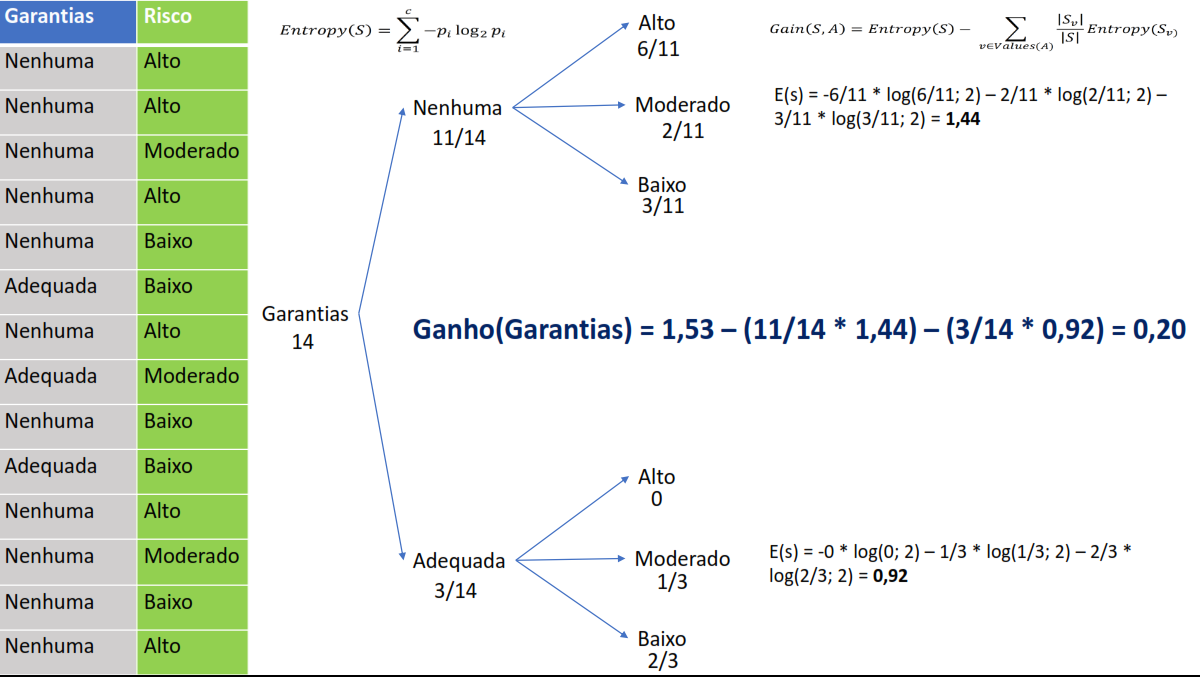


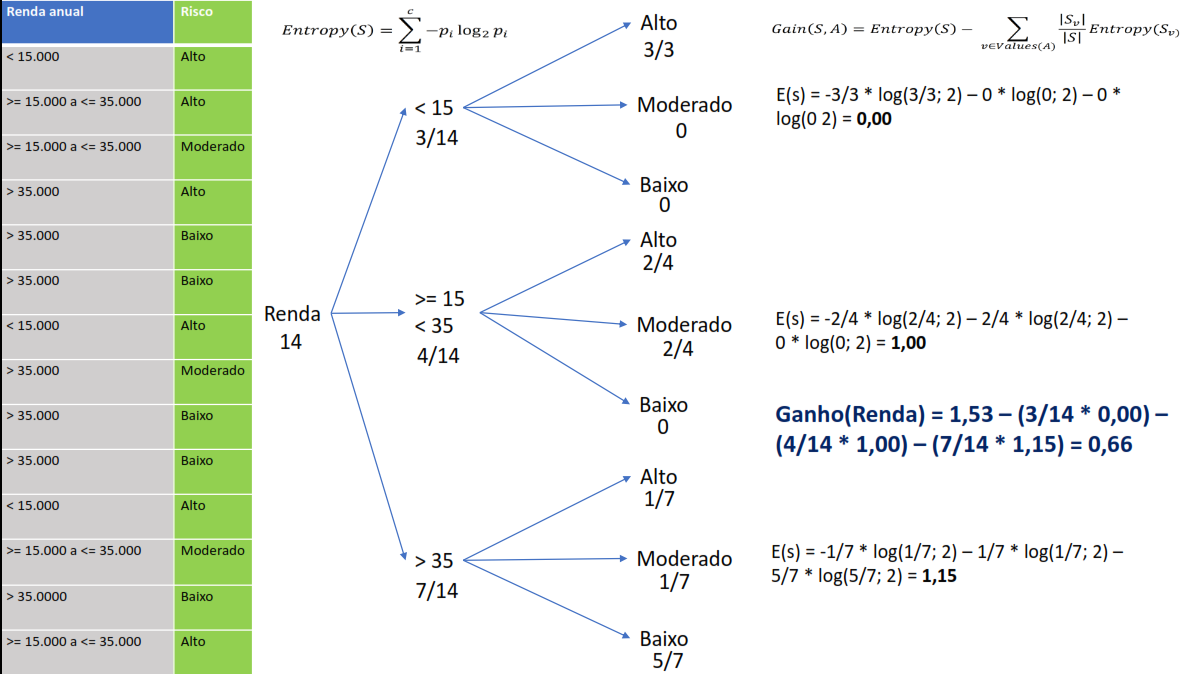
Quando todos em todos os nós a entropia for nula, o algoritmo para e obtêm-se a seguinte árvore de decisão:

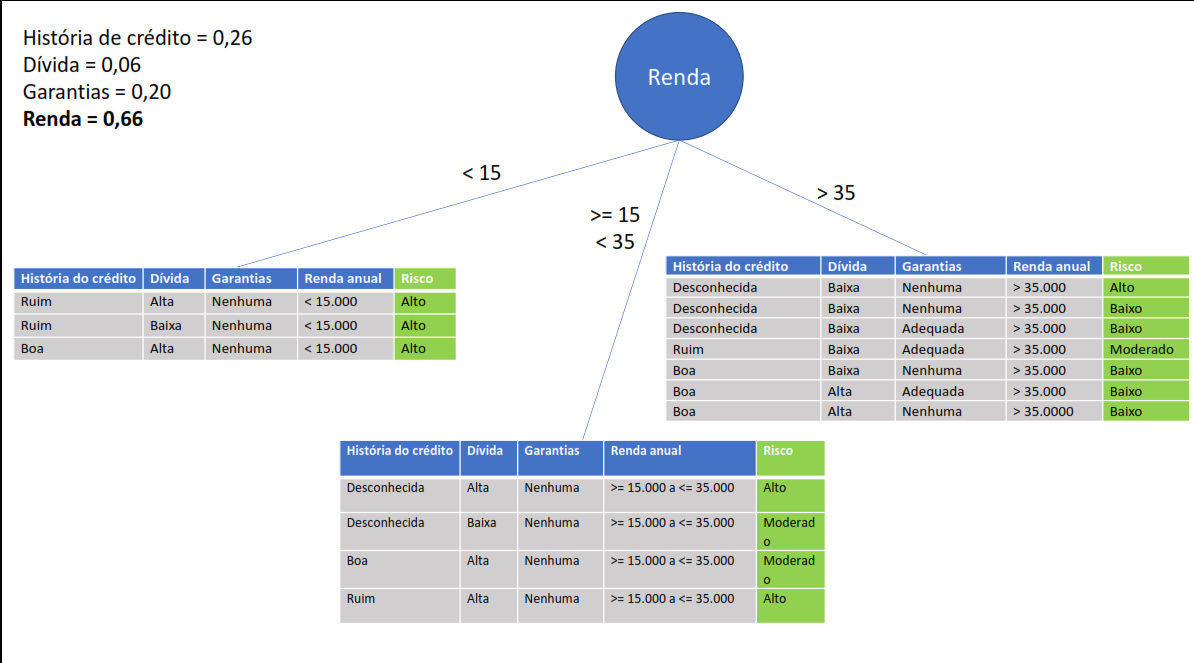


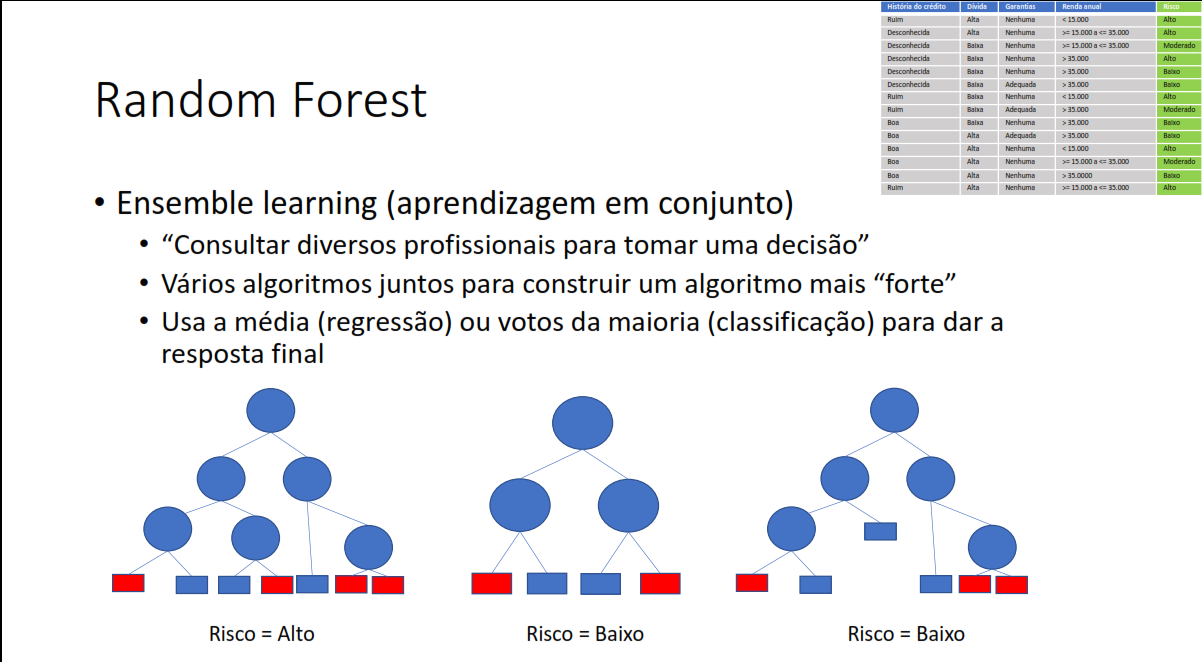


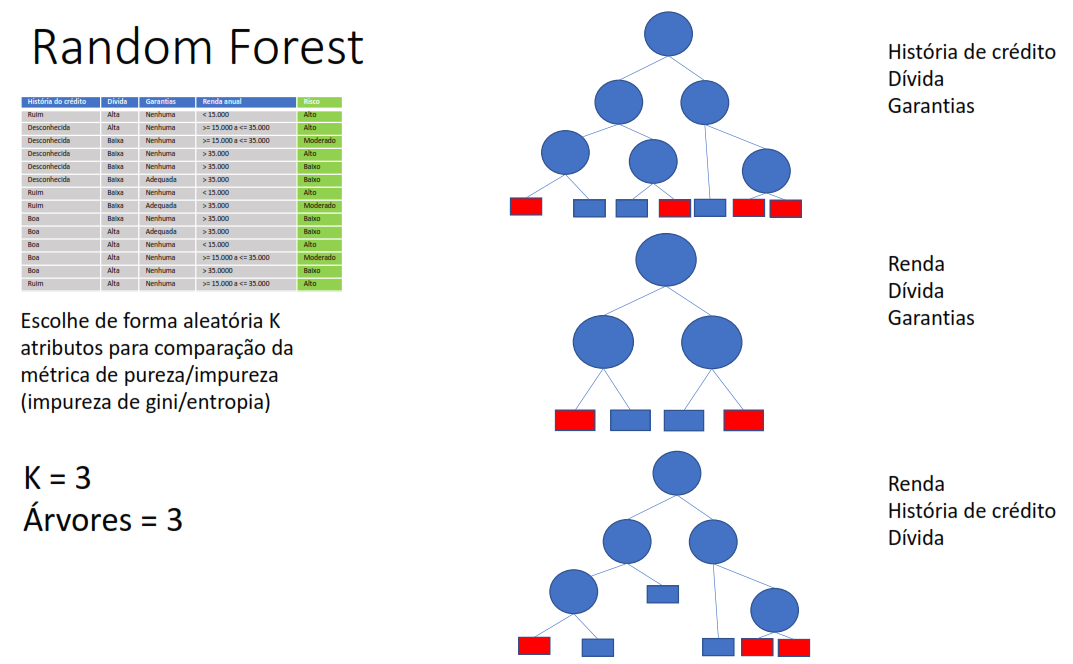












Comparação

**Árvore de decisão**: todo gerente de contratação possui um conjunto de critérios como nível de escolaridade, número de anos de experiência e desempenho da entrevista. Uma árvore de decisão é análoga a um gerente de contratação que entrevista candidatos com base em seus próprios critérios.

**Bagging**: Agora imagine em vez de um único entrevistador, agora existe um painel de entrevistas em que cada entrevistador tem um voto. **Bagging** ou **agregação** de bootstrap envolve a combinação de contribuições de todos os entrevistadores para a decisão final através de um processo democrático de votação.

**Floresta aleatória**: é um algoritmo baseado em Bagging com uma diferença chave, em que apenas um subconjunto de recursos é selecionado aleatoriamente. Em outras palavras, todo entrevistador só testará o entrevistado em determinadas qualificações selecionadas aleatoriamente (por exemplo, uma entrevista técnica para testar habilidades de programação e uma entrevista comportamental para avaliar habilidades não técnicas).

**Boosting**: Essa é uma abordagem alternativa em que cada entrevistador altera os critérios de avaliação com base no feedback do entrevistador anterior. Isso 'aumenta' a eficiência do processo de entrevista, implantando um processo de avaliação mais dinâmico.

**Gradient Boosting:** um caso especial de reforço em que os erros são minimizados pelo algoritmo de descida de gradiente, para as empresas de consultoria de estratégia aproveitam usando entrevistas de caso para eliminar candidatos menos qualificados.

**XGBoost**: pense no XGBoost como um Gradient Boosting com 'esteróides' (bem, é chamado de 'Extreme Gradient Boosting'). É uma combinação perfeita de técnicas de otimização de software e hardware para obter resultados superiores usando menos recursos de computação no menor período de tempo.

