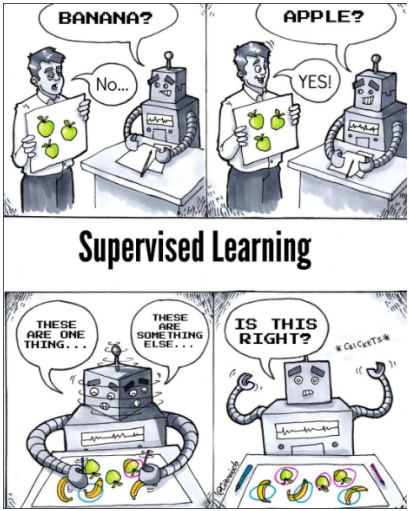
Parte IV: Aprendizado de Máquina.

Prof. Fabiano Araujo Soares, Dr. / FGA 0221 - Inteligência Artificial

Universidade de Brasília

2025

Aprendizado Supervisionado: Algoritmos



Aprendizado Supervisionado: Algoritmos

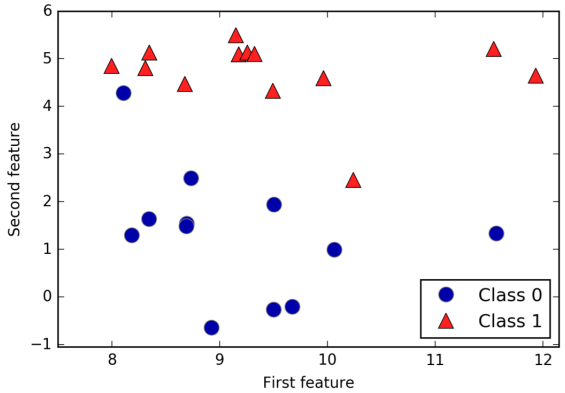
- A seguir temos a lista de alguns algoritmos utilizados em aprendizado supervisionado:
  - K-Nearest Neighbors
  - Modelos Lineares
  - Classificadores Bayesianos ingênuo
  - Arvores de Decisão
  - Essembles
  - Maquinas de Vetor de Suporte (SVM)
  - Redes Neurais (e Deep Learnig)

- Vamos explorar alguns deles.

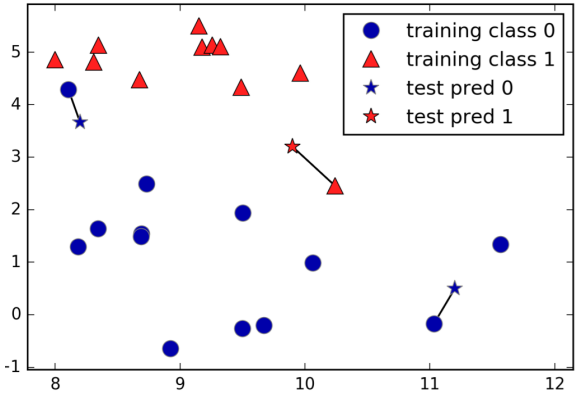
K-Nearest Neighbors (KNN)

- O algoritmo k-NN é o algoritmo de aprendizado de máquina mais simples;
- A construção do modelo consiste apenas em armazenar o conjunto de dados de treinamento;
- Para fazer uma previsão para uma nova amostra, o algoritmo encontra os pontos mais próximas no conjunto de dados de treinamento – seus "vizinhos mais próximos".

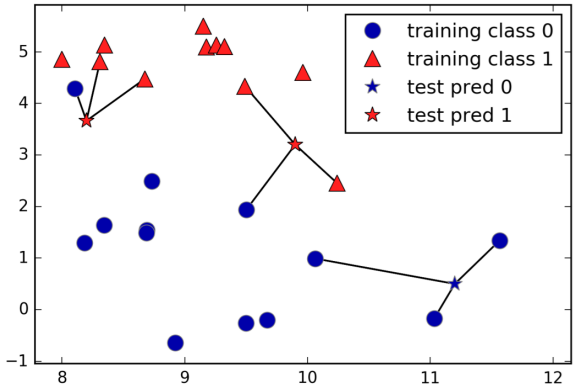
K-Nearest Neighbors (KNN)



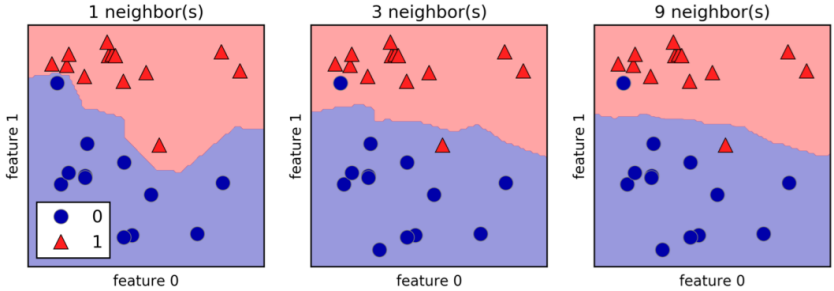
K-Nearest Neighbors (KNN)



K-Nearest Neighbors (KNN)



K-Nearest Neighbors (KNN)



K-Nearest Neighbors (KNN)

- Existem dois parâmetros importantes para o classificador K-NN: o número de vizinhos e como se mede a distância entre as amostras.
- O número de vizinhos utilizados depende da natureza dos dados, em geral, um pequeno número é mais adequado (três ou cinco);
- Sobre a medida de distância, uma possibilidade é a distância euclidiana;
- Um dos pontos fortes do k-NN é que o modelo é fácil de entender e muitas vezes tem um desempenho razoável sem muitos ajustes.
- Usar este algoritmo é um bom método para tentar antes de considerar técnicas mais avançadas.

K-Nearest Neighbors (KNN)

- Construir o modelo de vizinhos mais próximos geralmente é muito rápido;
- Quando seu conjunto de treinamento é muito grande (em número de características ou em número de amostras), a previsão pode ser lenta.
- Ao usar o algoritmo k-NN, é importante pré-processar seus dados.
- Essa abordagem geralmente não funciona bem em conjuntos de dados com muitos características (centenas ou mais);
- Seu desempenho é geralmente ruim com conjuntos de dados em que a maioria das características tem valor 0 na maioria das vezes (os chamados conjuntos de dados esparsos).

Modelos Lineares

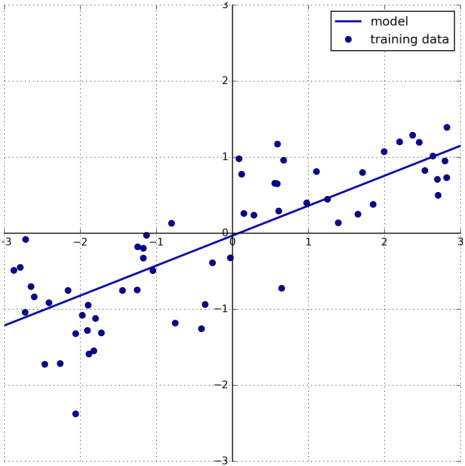
- Modelos lineares fazem uma previsão usando uma função linear das características de entrada;
- Os modelos lineares são da seguinte forma:

$$\hat{y} = \omega[0] \times x[0] + \omega[1] \times x[1] + ... + \omega[n] \times x[n] + b$$

- Onde $x[k]$ denota as características de uma amostra, ω e b são parâmetros dos modelos que são aprendidos e $\hat{y}$ é a predição que o modelo faz.
- A resposta prevista é uma soma ponderada das características de entrada, com pesos (que podem ser negativos) dados pelas entradas de ω.

Modelos Lineares

Se pensarmos em uma única característica, teremos uma regressão linear:



Modelos Lineares

- Regressão linear ou mínimos quadrados ordinários: encontra os parâmetros ω e b que minimize o erro quadrático médio entre as previsões e os verdadeiros alvos de regressão y, no conjunto de treinamento;
- O erro quadrático médio é a soma das diferenças quadráticas entre as previsões e os valores verdadeiros.
- A regressão linear não tem parâmetros, o que é um benefício, mas também não tem como controlar a complexidade do modelo.
- Com conjuntos de dados com mais dimensões, os modelos lineares se tornam mais poderosos.

Outros algoritmos de regressão:

- **Regressão de Ridge**: semelhante aos mínimos quadrados ordinários, mas os coeficientes w são escolhidos não apenas para que prevejam bem os dados de treinamento, mas também para que a magnitude dos coeficientes seja a menor possível evitando overfitting (Também conhecido como regularização L2).
- **Lasso**: Também restringe os coeficientes a serem próximos de zero, mas por regularização L1. Nesse caso, alguns coeficientes serão exatamente zero, ou seja, é como uma seleção de características automáticas. É interessante quando se tem um grande número de características.

Modelos Lineares para classificação

Vamos imaginar inicialmente uma classificação binária:

$$\hat{y} = \omega[0] \times x[0] + \omega[1] \times x[1] + ... + \omega[n] \times x[n] + b > 0$$

- Nesse caso, estamos traçando um limiar em "0";
- Se a função for menor que zero, classificamos a amostra como da classe -1, se a função for maior que zero, classificamos a amostra como da classe $+1$;
- Para modelos lineares para regressão, a saída, $\hat{y}$, é uma função linear das características: uma linha, plano ou hiperplano (em dimensões mais altas).

Modelos Lineares para classificação

Existem muitos algoritmos para aprender modelos lineares.

- Todos esses algoritmos diferem principalmente em:
  - Medição de quão bem ajustada é uma combinação de coeficientes (ω) e intercepto (b) ao conjunto de treinamento;
  - Que tipo de regularização eles usam.
- Os dois algoritmos de classificação linear mais comuns são:
  - regressão logística e
  - máquinas de vetor de suporte linear (SVMs lineares).

Modelos Lineares para classificação

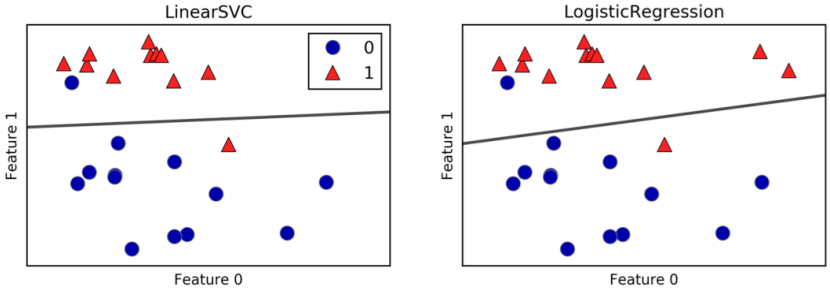


Figure 2-15. Decision boundaries of a linear SVM and logistic regression on the forge dataset with the default parameters

Modelos Lineares para classificação

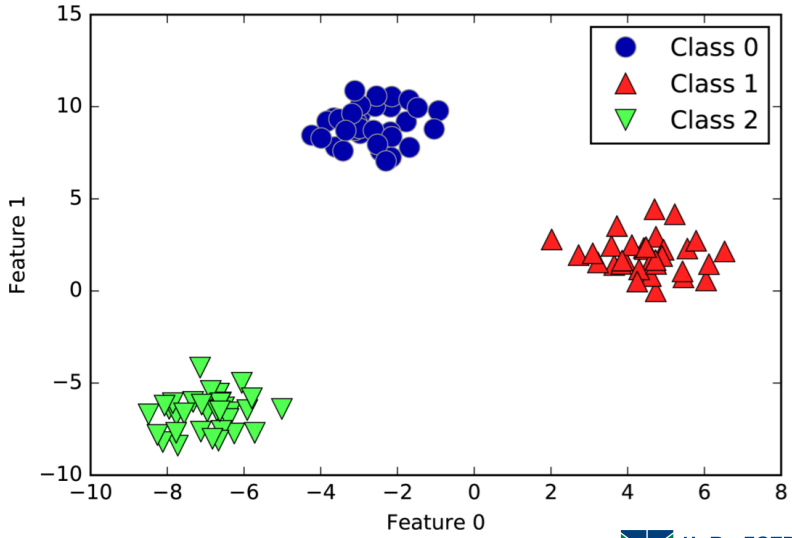
- Uma técnica comum para estender um algoritmo de classificação binária para um algoritmo de classificação multiclasse é a abordagem um contra todos;
- Na abordagem um contra todos, um modelo binário é aprendido para cada classe que tenta separar essa classe de todas as outras classes, resultando em tantos modelos binários quantas classes;
- Para fazer uma previsão, todos os classificadores binários são executados para uma amostra. O classificador que tiver a pontuação mais alta em sua classe "ganha" e esse rótulo de classe é retornado como a previsão.

Modelos Lineares para classificação

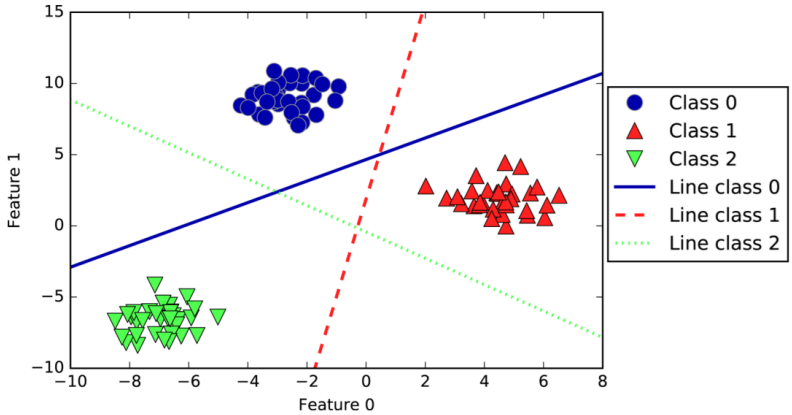
- Ter um classificador binário por classe resulta em ter um vetor de coeficientes (ω) e um intercepto (b) para cada classe;
- A classe para a qual a grandeza a seguir é mais alta é o rótulo de classe atribuída

$$\omega[0] \times x[0] + \omega[1] \times x[1] + ... + \omega[n] \times x[n] + b$$

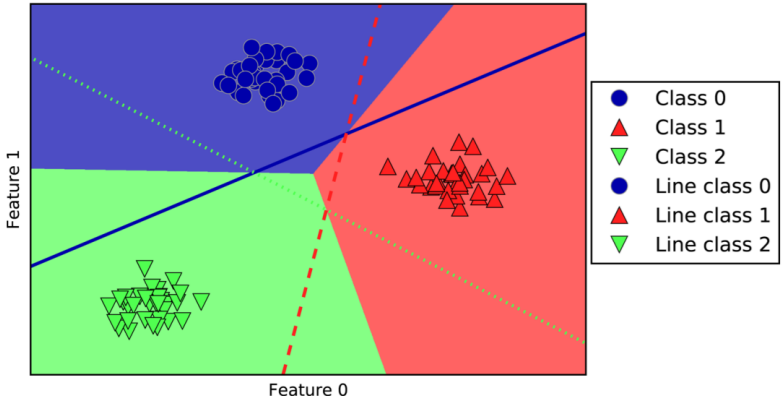
Modelos Lineares para classificação



Modelos Lineares para classificação



Modelos Lineares para classificação



Modelos Lineares para classificação

- O principal parâmetro dos modelos lineares é o parâmetro de regularização, denominado alpha nos modelos de regressão e C nos modelos lineares SVC e Logistic Regression;
- Valores grandes para alfa ou valores pequenos para C significam modelos simples;
- Se você assumir que apenas alguns de suas características são realmente importantes, você deve usar L1. Caso contrário, você deve usar como padrão L2;
- L1 também pode ser útil se a interpretabilidade do modelo for importante.

Modelos Lineares para classificação

- Modelos lineares são muito rápidos para treinar e também rápidos para fazer predições;
- Eles são dimensionados para conjuntos de dados muito grandes e funcionam bem com dados esparsos;
- Outro ponto forte dos modelos lineares é que eles tornam relativamente fácil entender como uma previsão é feita;
- Infelizmente, muitas vezes não é claro por que os coeficientes são do jeito que são. Isso é particularmente verdadeiro se seu conjunto de dados tiver características altamente correlacionadas.

Classificadores Bayesianos Ingênuos

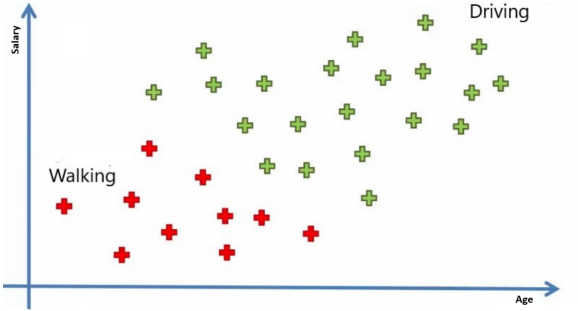
- Os classificadores Bayesianos Ingênuos são uma família de classificadores semelhantes aos classificadores por modelos lineares;
- No entanto, eles tendem a ser ainda mais rápidos no treinamento;
- O preço pago por essa eficiência é que os modelos ingênuos de Bayes geralmente fornecem desempenho de generalização piores do que o de classificadores lineares;
- A razão pela qual os modelos ingênuos de Bayes são tão eficientes é que eles aprendem parâmetros por examinar cada característica individualmente e coletar estatísticas simples por classe de cada característica.

Classificadores Bayesianos Ingênuos

- Existem três tipos mais comuns de classificadores Bayesianos ingênuos:
  - Gaussiano, que pode ser utilizado em dados contínuos;
  - Bernoulli, que assume dados binários;
  - Multinormal, que assume dados contáveis (isto é, que cada característica representa uma contagem inteira de alguns coisa, como a frequência com que uma palavra aparece em uma frase);
- Para fazer uma previsão, um ponto de dados é comparado com as estatísticas de cada uma das classes, e a melhor classe correspondente é prevista.
- Exemplo: https://www.analyticsvidhya.com/blog/2021/01/a-guide-to-the-naive-bayes-algorithm/

Exemplo: Classificados Bayesianos Ingênuo

Uma empresa está avaliando a probabilidade de um novo funcionário ir até o local de trabalho de carro ou a pé, baseado na idade e no salário. Os dados iniciais levantados estão no gráfico a baixo.

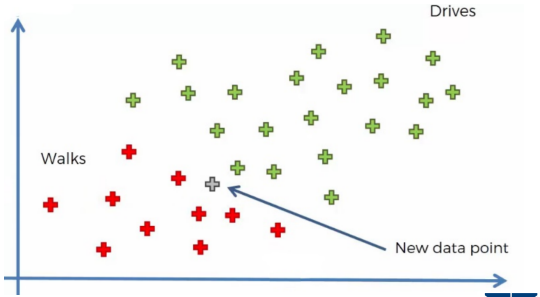


Exemplo: Classificados Bayesianos Ingênuo

Lembrando da equação de Bayes:

$$P(A \mid B) = \frac{P(B \mid A) \times P(A)}{P(B)}$$

Considerando agora um novo funcionário, qual é a classe (carro ou caminhando) mais provável para ele:



Exemplo: Classificados Bayesianos Ingênuo

Traduzindo o problema para a equação

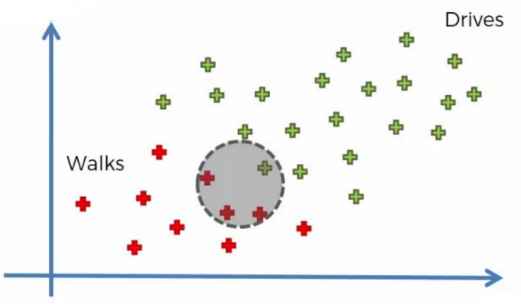
$$P(Walks \mid X) = \frac{P(X \mid Walks) \times P(Walks)}{P(X)}$$

Onde $P(Walks \mid X)$ é a probabilidade *a posteriori*, $P(X \mid Walks)$ é a probabilidade de verossimilhança, P(Walks) é a probabilidade *a priori* e P(X) é a verossimilhança marginal. Vamos levantar as probabilidades:

$$P(Walks) = \frac{\text{Number of Walkers}}{\text{Total Observations}} = \frac{10}{30}$$

Exemplo: Classificados Bayesianos Ingênuo

A fim de encontrar a verossimilhança marginal $P(X)$, temos que considerar um círculo ao redor do novo ponto de dados de qualquer raio, incluindo alguns pontos vermelhos e verdes.



$$P(X) = \frac{\text{Number of Similar Observations}}{\text{Total Observations}} = \frac{4}{30}$$

Exemplo: Classificados Bayesianos Ingênuo

Para encontrar $P(X \mid Walks)$ fazemos:

$$P(X \mid Walks) = \frac{\text{Number of Similar Observations Among those who Walk}}{\text{Total Observations}} = \frac{3}{10}$$

E finalmente, calculamos $P(Walks \mid X)$:

$$P(Walks \mid X) = \frac{\frac{3}{10} \times \frac{10}{30}}{\frac{4}{30}} = 0.75$$

Classificadores Bayesianos Ingênuos

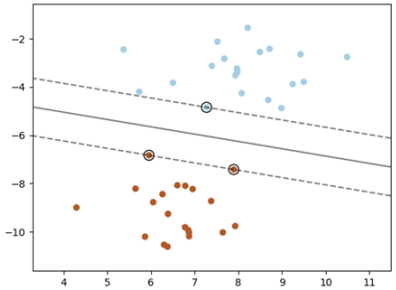
- Algoritmo de Bernoulli e Multinomial possuem apenas um parâmetro alfa, que controla a complexidade do modelo;
- Um alfa grande significa uma suavização maior nos modelos estatísticos (modelos mais simples);
- O desempenho do algoritmo é relativamente robusto a configuração de alfa, o que significa que a configuração de alfa não é crítica para um bom desempenho. No entanto, ajustá-lo geralmente melhora a precisão.

Classificadores Bayesianos Ingênuos

- Classificadores Gaussianos são usados principalmente em dados com um número grande de dimensões;
- Classificadores de Bernoulli e Multinomial são usados principalmente para dados esparsos (como texto);
- Os modelos ingênuos de Bayes compartilham muitos dos pontos fortes e fracos dos modelos lineares:
  - Eles são muito rápidos para treinar e fazer previsões;
  - O procedimento de treinamento é fácil para entender;
  - Os modelos funcionam muito bem com dados esparsos, de alta dimensão e são relativamente robustos aos parâmetros.
- Os modelos Bayesianos ingênuos são frequentemente usados em conjuntos de dados muito grandes, onde o treinamento até mesmo de um modelo linear pode ser longo.

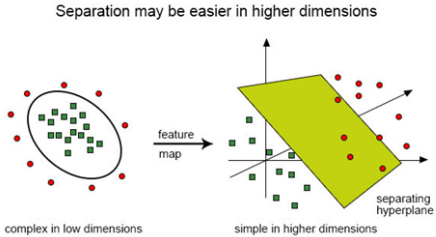
Máquinas de Vetores de Suporte (SVM)

- Algoritmo de aprendizado supervisionado para classificação e regressão.
- O objetivo é encontrar o hiperplano que maximiza a margem entre diferentes classes.
- Eficaz para casos em que os dados não são linearmente separáveis (uso do "truque do kernel").



Como o SVM Funciona?

- Dados de entrada são representados em um espaço N-dimensional.
- O SVM determina o hiperplano ótimo que maximiza a distância (margem) entre os exemplos mais próximos das diferentes classes (vetores de suporte).
- Casos não linearmente separáveis usam funções kernel para transformar os dados.



SVM: Equação do Hiperplano

Hiperplano de Separação

Dados x são classificados por:

$$w \cdot x + b = 0$$

onde:

- w: vetor de pesos
- b: intercepto

SVM: Equação do Hiperplano

Maximizando a Margem

O SVM busca:

$$\max \frac{2}{\|w\|}$$

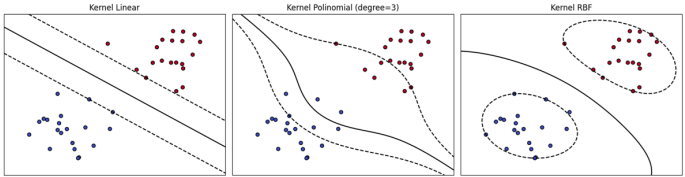
sujeito a $y_i(w \cdot x_i + b) \geq 1$ para todo i.
Onde $\|w\|$ é a norma do vetor w e é caclulada fazendo:

$$\|w\| = \sqrt{w_1^2 + w_2^2 + ... + w_n^2}$$

para um vetor w de dimensão n.

Kernel Trick no SVM

- Para dados não linearmente separáveis, o SVM usa funções kernel para projetar os dados em espaços de maior dimensão.
- Exemplos comuns de kernel:
  - Linear
  - Polinomial
  - Radial Basis Function (RBF)



Random Forest: Algoritmo de Floresta Aleatória

- Algoritmo de aprendizado supervisionado para classificação e regressão.
- Constrói múltiplas árvores de decisão (ensembling).
- Cada árvore é treinada em uma amostra aleatória (bootstrap) do conjunto de dados.
- Resultado final é a combinação (votação na classificação ou média na regressão) das árvores individuais.

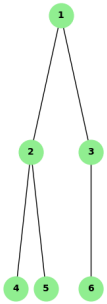
Como Funciona o Random Forest?

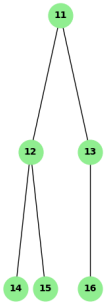
1. Para cada árvore do modelo:
   - Seleciona uma amostra bootstrap do conjunto de treinamento.
   - Em cada divisão do nó, escolhe um subconjunto aleatório das variáveis.
   - A divisão é selecionada com base na melhor separação disponível no subconjunto.
   - A árvore cresce até um tamanho pré-definido ou critério de parada.
2. O modelo final agrega as previsões de todas as árvores para formar a decisão.

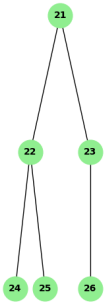
Vantagens do Random Forest

- Reduz o overfitting típico de uma única árvore de decisão.
- Robustez a ruídos e dados faltantes.
- Pode lidar bem com variáveis categóricas e contínuas.
- Avaliação automática da importância das variáveis.

Visualização Conceitual







Cada árvore é construída com diferentes amostras e subconjuntos de atributos, aumentando a diversidade e a robustez do modelo.

Conclusão

- Aprendizado supervisionado é uma abordagem fundamental em IA e Machine Learning.
- Revisamos algoritmos importantes como KNN, modelos lineares, Naive Bayes, SVM e Random Forest.
- Cada algoritmo tem vantagens e aplicações específicas, sendo importante entender suas características.
- Técnicas como regularização e kernel trick ampliam a capacidade dos modelos.
- Compreender esses métodos é essencial para avançar em tópicos mais complexos, como deep learning.

Próxima Aula: Redes Neurais com Treinamento Supervisionado

- Introdução às redes neurais artificiais.
- Arquiteturas básicas e funcionamento.
- Treinamento supervisionado: optimização por backpropagation.
- Exemplos práticos e aplicações.
- Preparação para aprofundamento em deep learning.

Referências

- Russell, S., Norvig, P., "Artificial Intelligence: A Modern Approach", 4th ed., Person, 2022.
- Duda, Richard O., "Pattern Classification", 2nd ed., Wiley, 2000.
- Muller, A. C., Guido, S., "Introduction to Machine Learning with Python A Guide for Data Scientists", O'Reilly, 2017.

Obrigado!

E-mail: fabianosoares@unb.br

LinkedIn: https://www.linkedin.com/in/fabiano-soares-06b6a821a/

Site do curso: https://www.fabianosoares.eng.br/fga0221-inteligência-artificial