Aprendizado Supervisionado

Definição

- Algoritmos que usam **dados rotulados** (entrada e saída conhecida) para aprender a mapear novos dados e fazer previsões ou classificações;

- O objetivo é permitir que o modelo **generalize os padrões aprendidos para fazer previsões precisas em dados novos e não vistos**.

- Classificação de e-mails como spam ou não spam, previsão de preços de imóveis, diagnóstico médico, etc.

Classificação

- Envolve **atribuir rótulos a diferentes categorias ou classes** com base nas características dos dados;

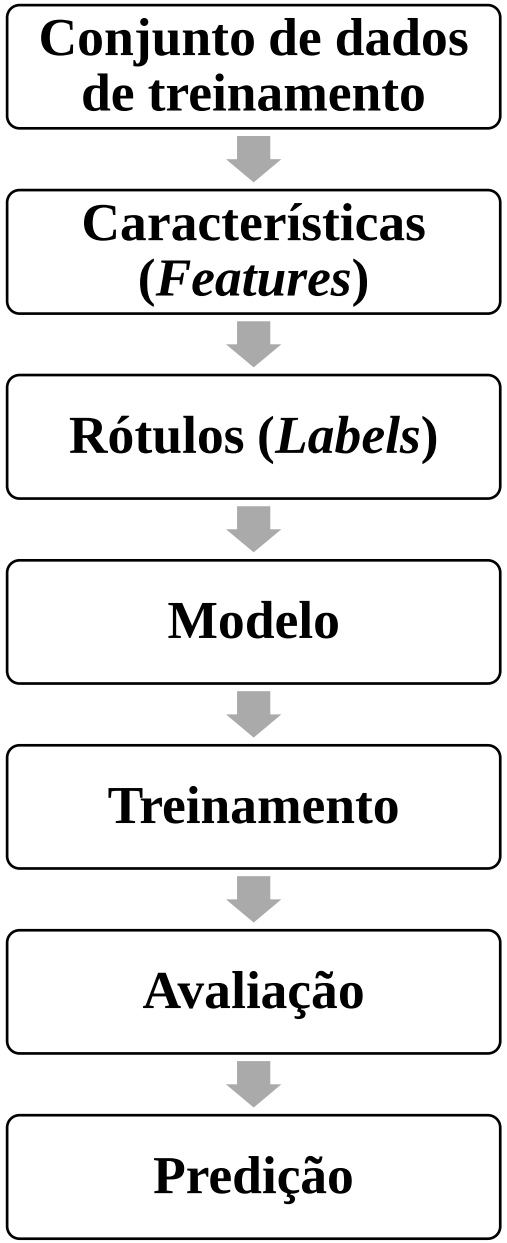
- Exemplo, determinar se um e-mail é spam ou não spam, identificar se uma imagem contém um gato ou um cachorro, ou prever se um paciente tem uma doença específica ou não.

Regressão

- Envolve **prever valores numéricos contínuos** com base nas características dos dados;

- Exemplo: prever o preço de uma casa com base em suas características (como número de quartos, área, etc.), prever a quantidade de vendas de um produto com base em vários fatores, ou prever a temperatura com base em informações climáticas.

Componentes do Aprendizado Supervisionado


Conjunto de dados de treinamento
Características (*Features*)
Rótulos (*Labels*)
Modelo
Treinamento
Avaliação
Predição

Conjunto de Dados

- É o conjunto de dados utilizado para treinar o modelo;

- Cada exemplo no conjunto de dados possui características (variáveis independentes) e um rótulo (variável dependente) associado.

Características (*Features*)

- São as variáveis independentes que descrevem cada exemplo;

- Exemplo: na construção de um modelo para prever o preço de casas, as características podem incluir tamanho, número de quartos, localização, etc.

Rótulos (*Labels*)

- São as saídas desejadas que o modelo deve prever;

- No caso do exemplo das casas, o rótulo seria o preço real da casa.

Modelo

- É o algoritmo que é treinado com os dados de treinamento para fazer previsões;

- O modelo aprenderá a relação entre as características e os rótulos durante o treinamento.

Treinamento

- É o processo de ajustar os parâmetros do modelo usando o conjunto de dados de treinamento;

- O objetivo é fazer com que o modelo aprenda a melhor maneira de mapear as características nos rótulos.

Avaliação

- Após o treinamento, o modelo é avaliado usando um conjunto separado de dados chamado conjunto de dados de teste;

- Isso permite verificar como o modelo se comporta em dados não vistos.

Predição

- Uma vez que o modelo é treinado e avaliado, ele pode ser usado para fazer previsões em novos dados, onde os rótulos são desconhecidos.

Algoritmos

Algoritmos de classificação

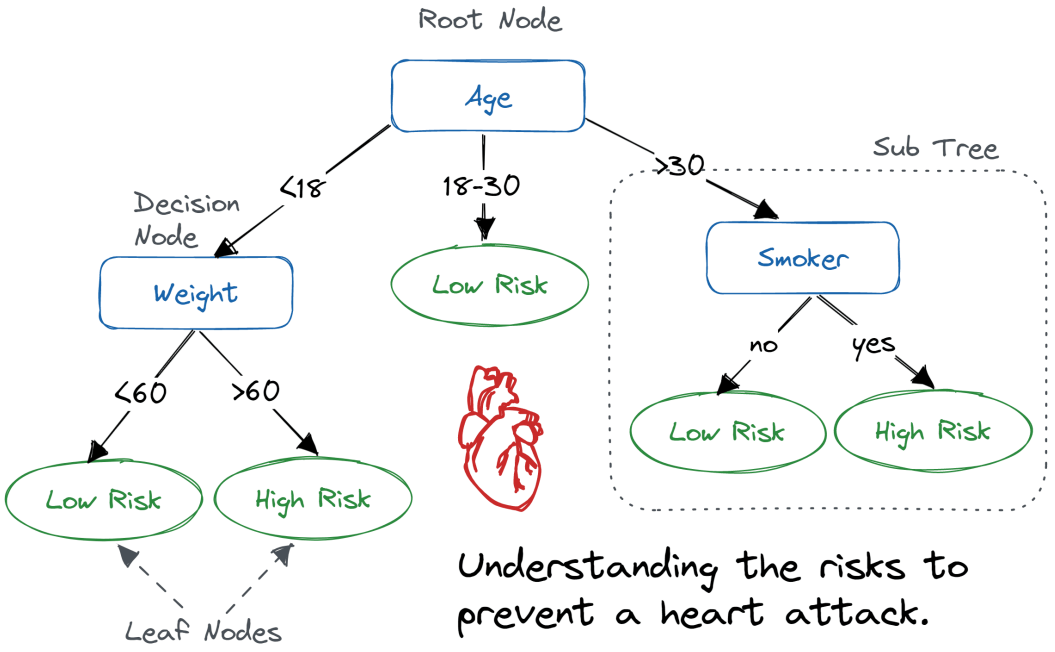
Algoritmo	Descrição	Exemplos de Implementação
Árvores de Decisão	Estrutura de árvore para tomada de decisões	*C4.5, CART, Random Forest*
K-Nearest Neighbors (KNN)	Classificação baseada na maioria dos vizinhos mais próximos	*KNeighborsClassifier*
Naive Bayes	Baseado no Teorema de Bayes, considera independência entre características	*GaussianNB, MultinomialNB*
Support Vector Machines (SVM)	Encontra um hiperplano de separação otimizado	SVC
Regressão Logística	Modelo probabilístico usando função logística	*LogisticRegression*
Gradient Boosting	Combinação sequencial de modelos fracos	*GradientBoostingClassifier, XGBClassifier, LightGBM, CatBoost*
Redes Neurais Artificiais	Modelos inspirados em redes neurais do cérebro	*MLPClassifier*

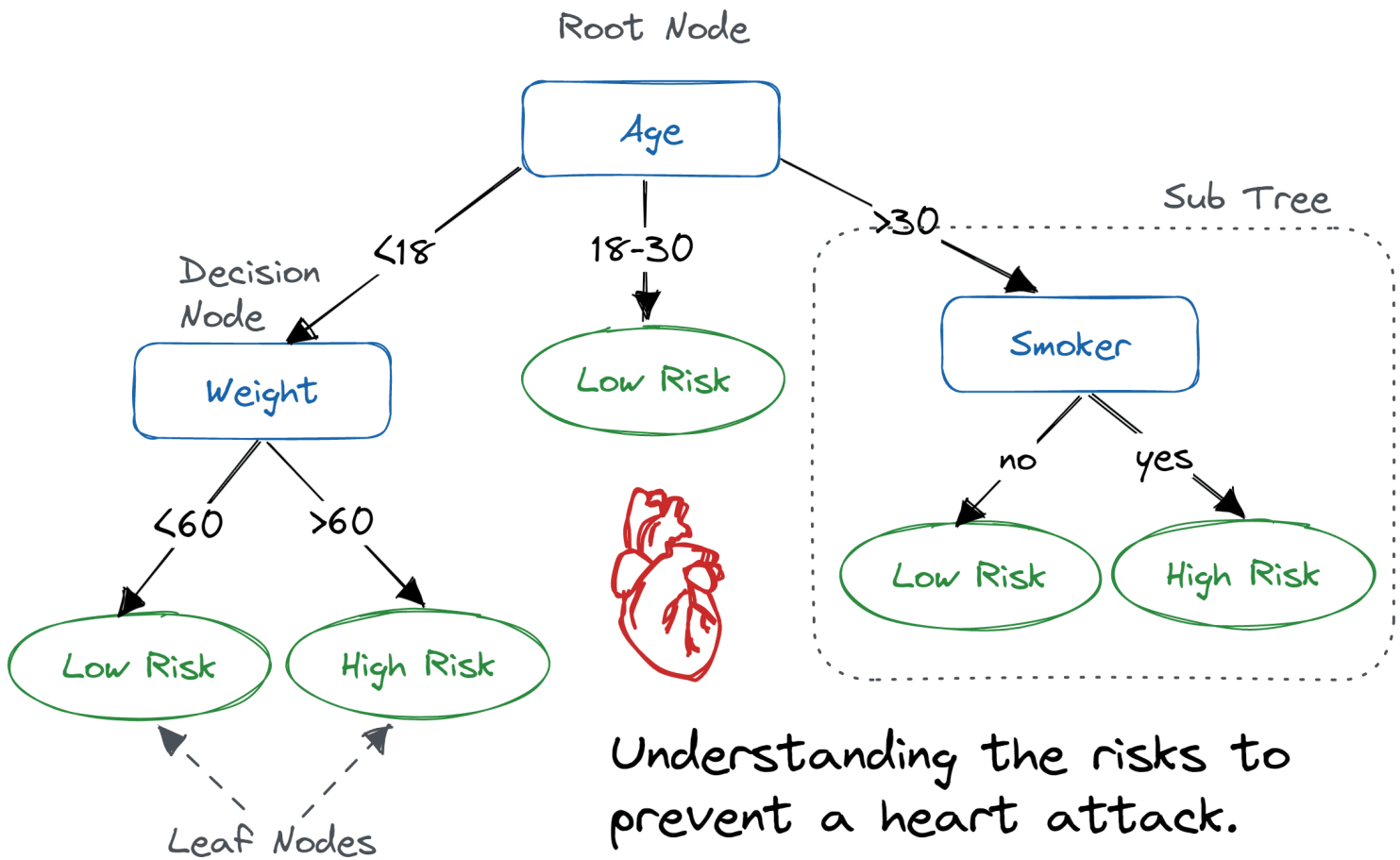
Árvores de Decisão

- Representam uma série de decisões e suas possíveis consequências em forma de uma estrutura de árvore;

- Cada nó interno representa uma decisão baseada em uma característica, enquanto as folhas representam os resultados finais, como uma classe ou um valor;

- Podem ser empregadas tanto para classificação quanto para regressão.





Understanding the risks to prevent a heart attack.

C4.5

- Utiliza a métrica de ganho de informação para determinar a melhor característica para dividir os dados em cada nó da árvore;

- É eficaz para dados categóricos e também suporta valores faltantes;

- Constrói árvores completas e, em seguida, poda-as para evitar o sobreajuste *(overfitting).*

CART (*Classification and Regression Trees*)

- É utilizado tanto para classificação quanto para regressão;

- Ao contrário do C4.5, o CART utiliza o índice de Gini para medir a impureza dos dados e escolher as divisões;

- Constrói árvores completas e utiliza um processo de poda para evitar o sobreajuste *(overfitting)*.

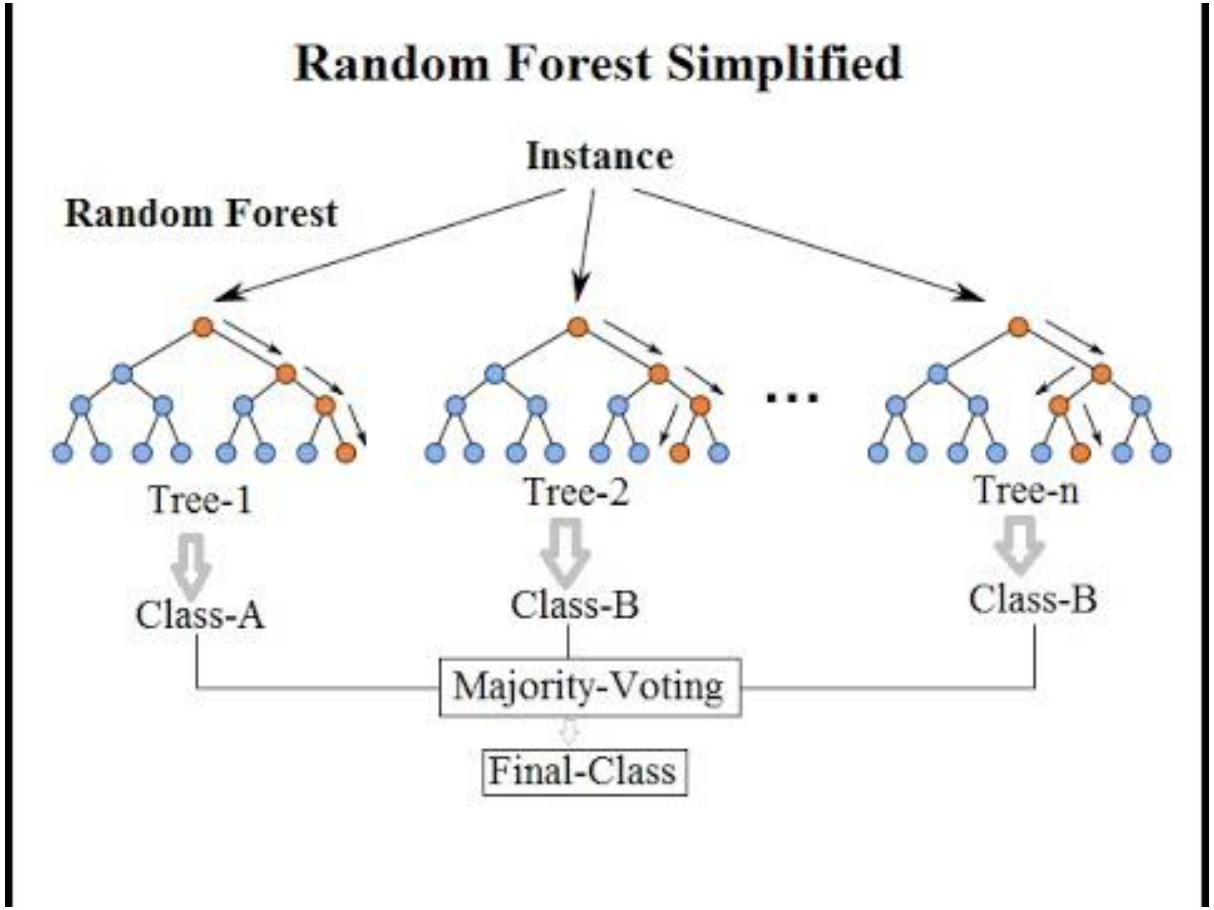
Random Forest

- É uma extensão das árvores de decisão que constrói várias árvores em paralelo e combina suas previsões para tomar uma decisão final;

- Cada árvore é construída em um subconjunto aleatório dos dados e usa uma parte aleatória das características, evitando o sobreajuste e aumentando a precisão;

- No final, as previsões das árvores individuais são agregadas para produzir uma previsão final mais robusta.

Random Forest



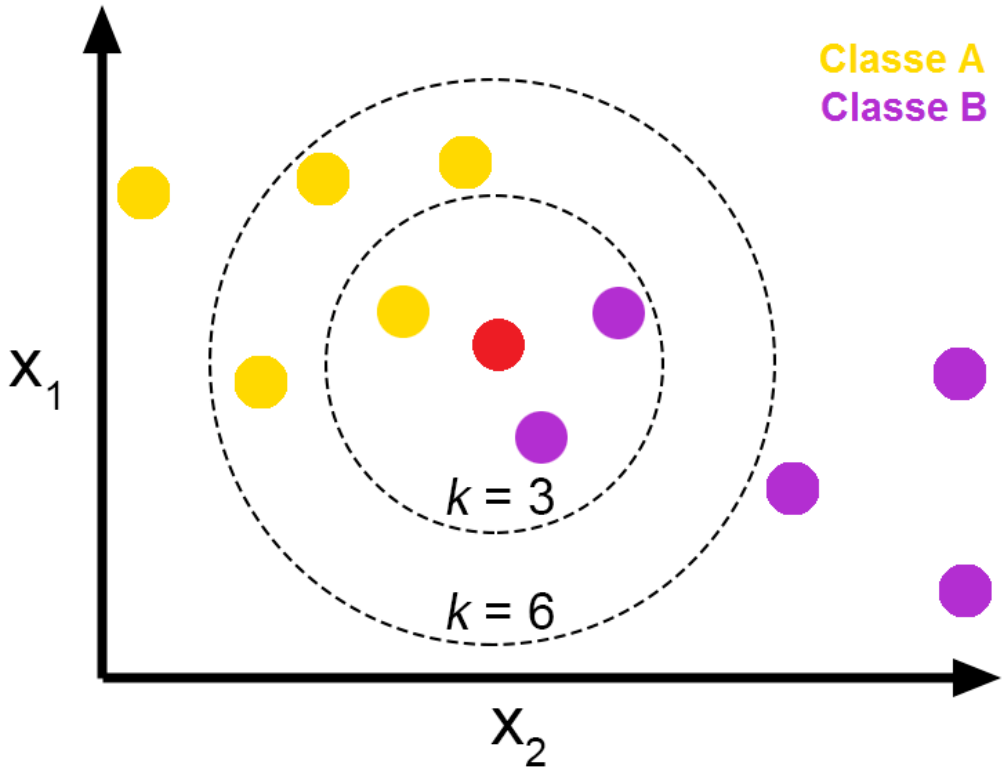
Árvores de Decisão -Resumo

Algoritmo	**Métrica de Divisão**	**Valores Ausentes**	**Poda da Árvore**	**Dados Numéricos**	**Dados Categóricos**	***Overfitting***	***Ensemble***
C4.5	Ganho de Informação	Suportado	Sim	Suportado	Suportado	Sim	Não
CART	Índice de Gini	Suportado	Sim	Suportado	Suportado	Sim	Não
Random Forest	Índice de Gini / Ganho de Informação	Suportado	Não	Suportado	Suportado	Menor	Sim

K-Nearest Neighbors (KNN)

- Baseia-se no princípio de que instâncias semelhantes tendem a ter rótulos semelhantes;

- O KNN classifica um novo ponto de dados com base nos rótulos dos pontos de dados vizinhos mais próximos a ele.

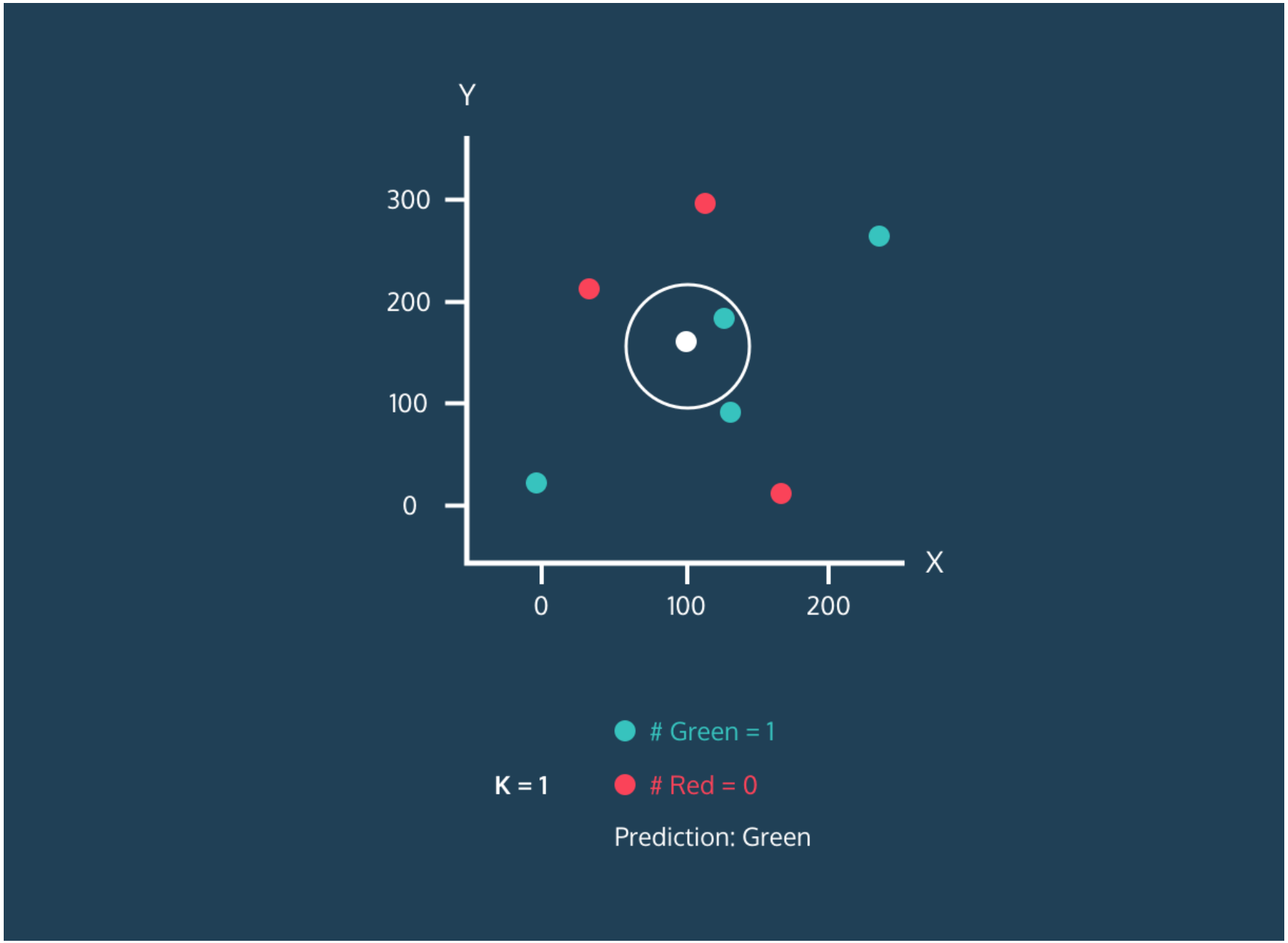


1. **Escolha do Parâmetro K:** O primeiro passo é escolher o valor de K, que representa o número de vizinhos mais próximos que serão considerados para tomar uma decisão de classificação ou regressão;

2. **Medição de Distância:** O KNN utiliza uma métrica de distância (geralmente a distância euclidiana) para calcular a proximidade entre os pontos de dados. Quanto mais próximos os pontos de dados estiverem, mais semelhantes eles são considerados;

3. **Encontrar Vizinhos Mais Próximos:** O algoritmo identifica os K pontos de dados mais próximos ao ponto de dados de teste com base na métrica de distância escolhida;

4. **Votação (Classificação) ou Média (Regressão):** Para classificação, os rótulos dos K vizinhos mais próximos são usados para determinar a classe do ponto de dados de teste. O rótulo mais frequente entre os vizinhos é atribuído ao ponto de dados de teste. Para regressão, a média dos valores alvo dos K vizinhos é calculada e usada como a previsão para o ponto de dados de teste.


Y
300
200
100
0
X
0
100
200
# Green = 1
K = 1
# Red = 0
Prediction: Green

Naive Bayes

- Método baseado no teorema de Bayes, que é uma técnica estatística para calcular a probabilidade de um evento ocorrer com base em evidências prévias relacionadas a esse evento;
- O termo "*Naive*" (ingênuo) refere-se à suposição de independência condicional entre as características (variáveis) do conjunto de dados, o que simplifica o cálculo das probabilidades;

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

1. **Conjunto de Treinamento:** O algoritmo Naive Bayes requer um conjunto de treinamento que consiste em instâncias rotuladas com suas características;

2. **Cálculo de Probabilidades:** Para cada classe possível, o Naive Bayes calcula a probabilidade de um ponto de dados pertencer a essa classe, com base nas probabilidades das características dadas essa classe. Isso é feito usando o teorema de Bayes;

3. **Classificação:** Para classificar um novo ponto de dados, o algoritmo calcula as probabilidades de pertencer a cada classe possível e atribui o rótulo da classe com a maior probabilidade.

Naive Bayes Gaussiano: variante usada quando as características são contínuas e seguem uma distribuição gaussiana (normal). Assume que as características de cada classe têm uma distribuição normal;

Naive Bayes Multinomial: empregado quando as características são discretas e representam a contagem de ocorrências de certos eventos. É frequentemente usado para tarefas envolvendo dados de texto, como classificação de documentos, análise de sentimentos e categorização de notícias;

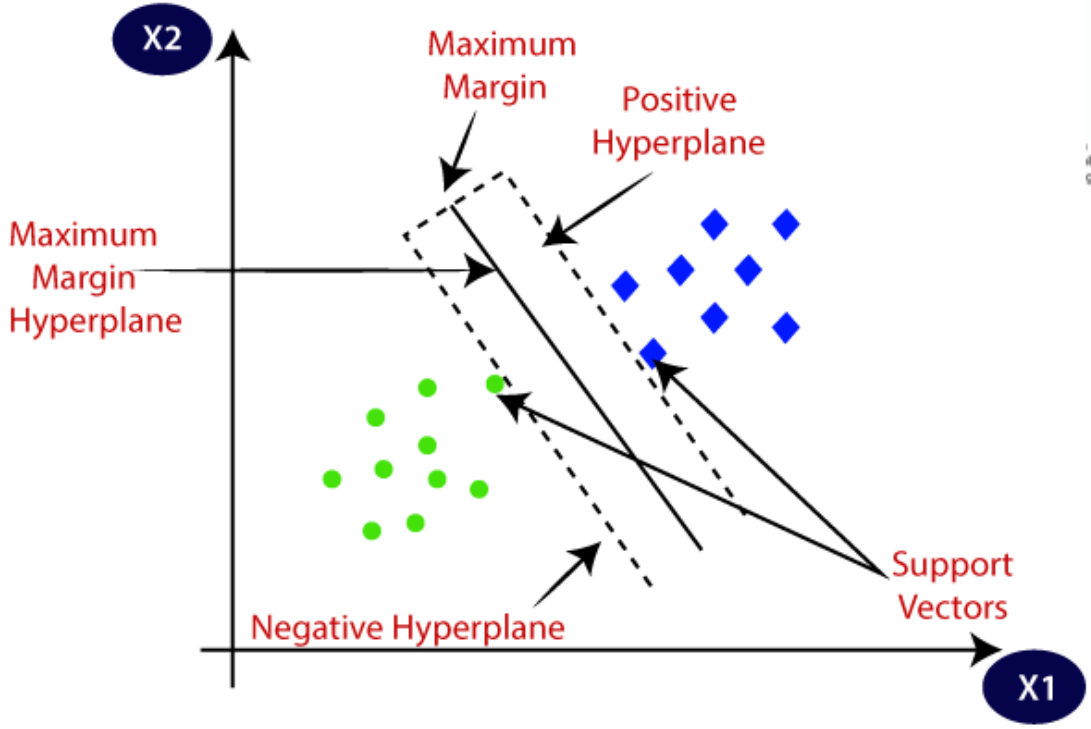
Naive Bayes Bernoulli: Semelhante ao Naive Bayes Multinomial, mas é adequada para dados binários, onde cada característica pode estar presente ou ausente. É útil para problemas em que você deseja prever se algo ocorrerá ou não, como detecção de spam ou análise de presença/ausência de palavras em um texto.

Support Vector Machines (SVM)

- As Máquinas de Vetores de Suporte (SVM) são um algoritmo de aprendizado de máquina usado principalmente para tarefas de classificação e regressão;

- São especialmente eficazes em cenários de classificação, onde os dados são complexos e não linearmente separáveis;

- A principal ideia por trás das SVMs é encontrar um hiperplano no espaço de características que melhor separe as classes de interesse.



1.Princípio da Separação Ótima: O objetivo das SVMs é encontrar o hiperplano que separa as classes de dados de maneira que a margem entre as classes seja a mais ampla possível. A margem é a distância entre o hiperplano e os pontos de dados mais próximos de cada classe. Esse hiperplano é chamado de hiperplano de separação;

2. Margens e Vetores de Suporte: Os pontos de dados que estão mais próximos do hiperplano de separação são chamados de "vetores de suporte". As SVMs focam nesses vetores de suporte para determinar a posição e a orientação do hiperplano. A escolha do hiperplano de separação é influenciada pela posição dos vetores de suporte;

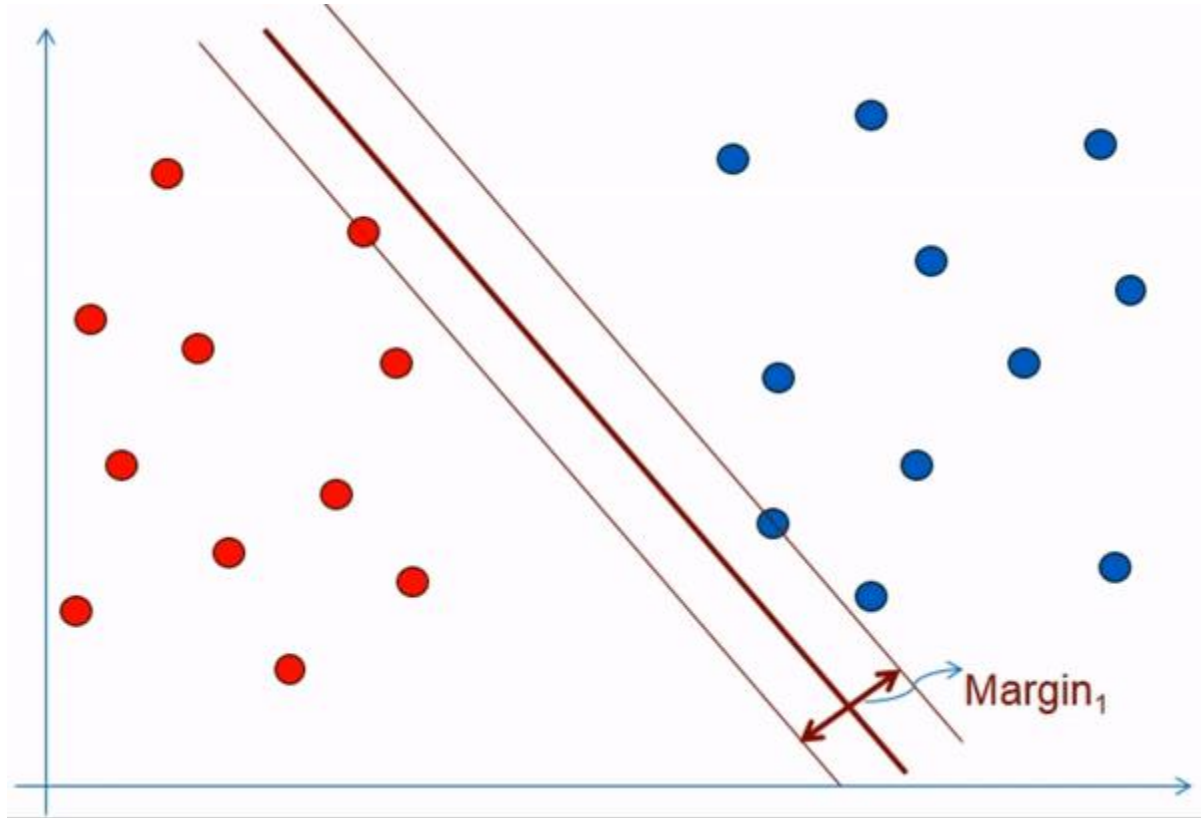
***3.Kernel Trick*:** Em casos onde os dados não são linearmente separáveis no espaço de características original, as SVMs podem usar uma função de kernel para mapear os dados para um espaço de características de dimensão superior onde a separação é possível. Isso permite que as SVMs lidem com dados complexos e não lineares;

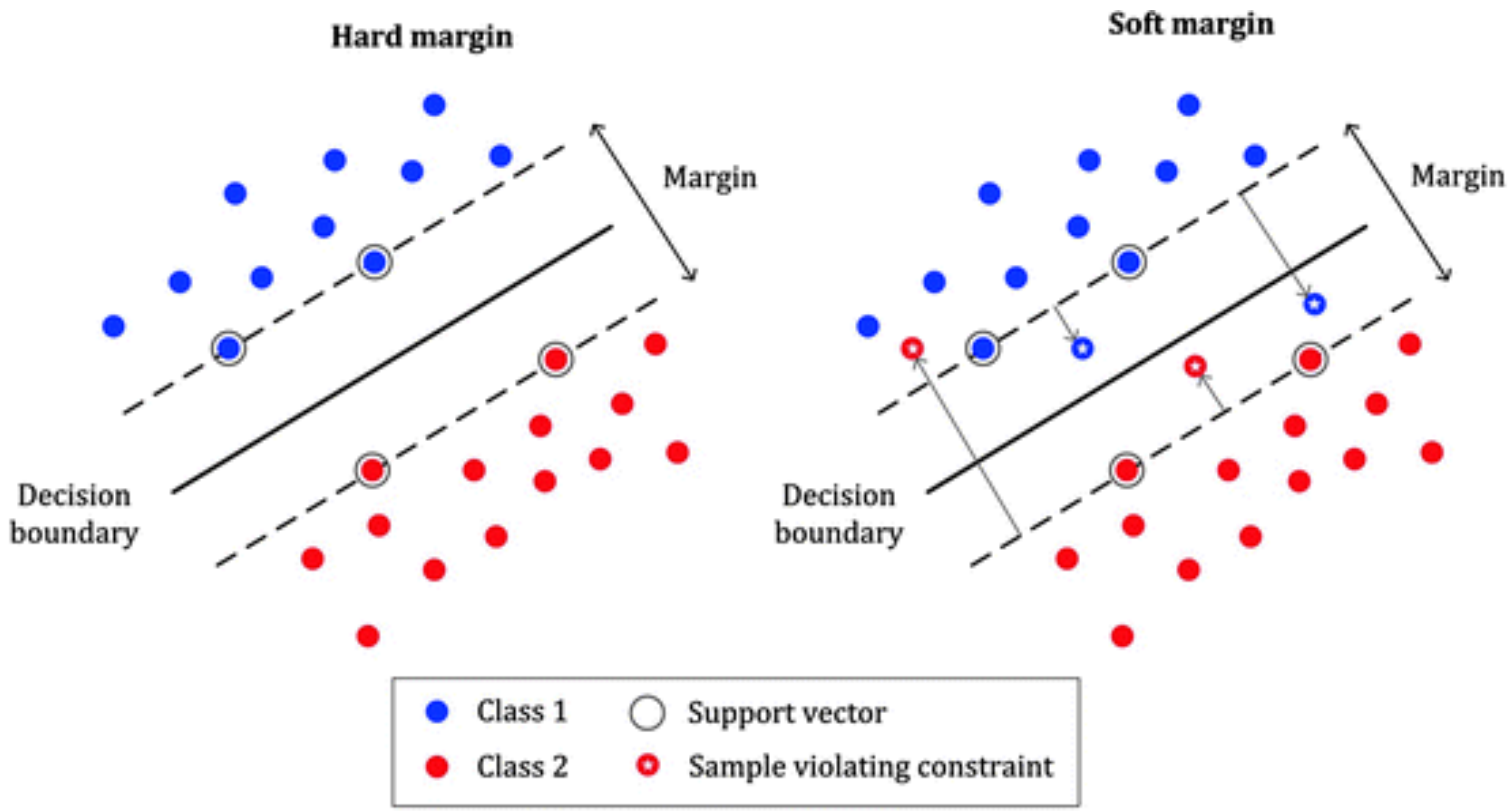
4.Classificação e Regressão: As SVMs são comumente usadas para tarefas de classificação, onde a meta é separar as amostras em diferentes classes. No entanto, as SVMs também podem ser usadas para regressão, onde o objetivo é encontrar uma curva que melhor se ajusta aos dados.

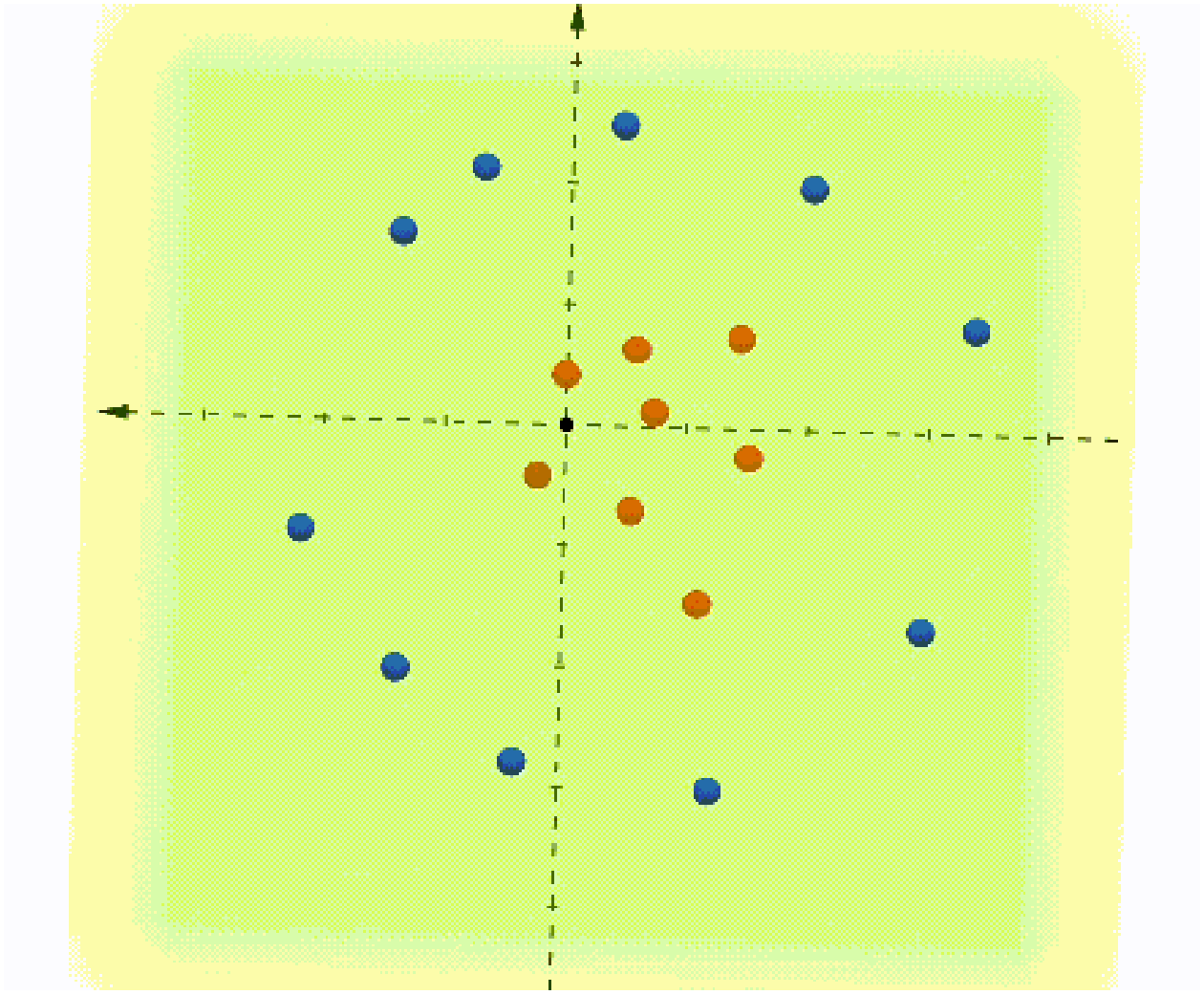
5. **Custo de Regularização:** As SVMs introduzem o conceito de um termo de regularização, que controla a importância de minimizar a margem em relação a classificar corretamente os pontos. Esse termo é importante para evitar o *overfitting*;

6. **SVM de margem suave:** Em alguns casos, os dados podem não ser completamente separáveis com uma margem rígida. Um SVM com margem suave permite uma margem de erro ao classificar os pontos, permitindo que alguns pontos fiquem no lado errado do hiperplano;

7. **Várias Classes:** Embora as SVMs sejam originalmente projetadas para problemas de duas classes, elas podem ser estendidas para tarefas de várias classes usando abordagens como o "*One-vs-Rest*" ou "*One-vs-One*“.


Margin$_1$


Hard margin
Margin
Decision boundary
Soft margin
Margin
Decision boundary
Class 1
Class 2
Support vector
Sample violating constraint

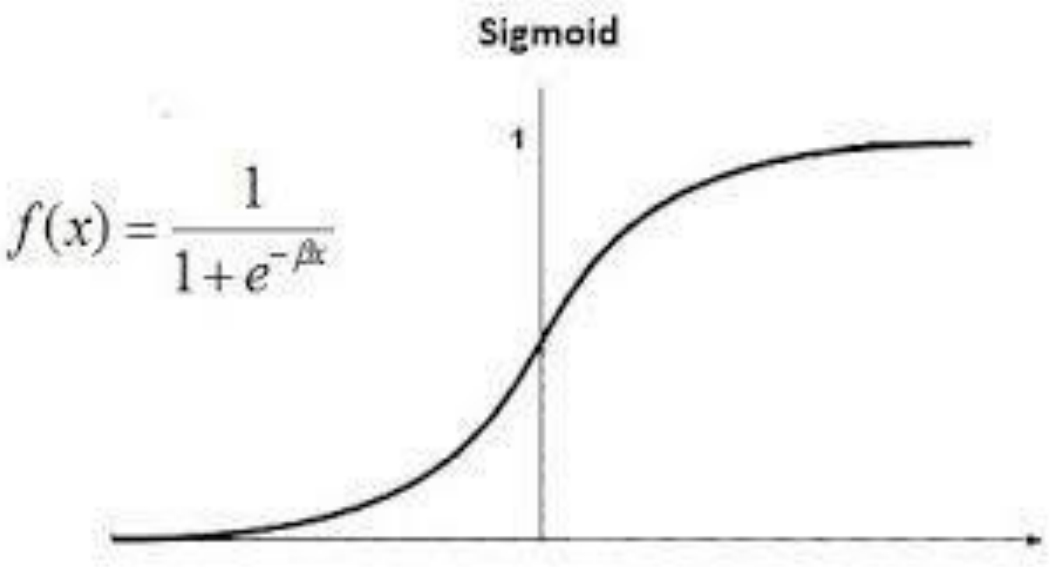


Regressão Logística

- Usado para tarefas de classificação binária e, com algumas modificações, também pode ser usada para classificação multiclasse;

- Apesar do nome, a Regressão Logística é usada para tarefas de classificação, não para regressão;

- É simples de implementar, interpretar e entender. Funciona bem quando as classes são linearmente separáveis ou quase separáveis.



Função Logística (Sigmoide): utiliza a função logística (ou função sigmoide) para modelar a probabilidade de uma amostra pertencer a uma determinada classe. A função logística transforma qualquer valor real em um valor entre 0 e 1, que pode ser interpretado como uma probabilidade;

Modelo Linear Generalizado: é um exemplo de um Modelo Linear Generalizado (GLM), que é uma generalização dos modelos de regressão linear. Ele estende a regressão linear para problemas de classificação, permitindo que a saída do modelo esteja na forma de probabilidades;

Função de Decisão: calcula uma função de decisão que estima a probabilidade de uma amostra pertencer à classe positiva (ou à classe "1" em uma tarefa binária). Se a probabilidade calculada for maior que um determinado limiar, a amostra é classificada como pertencente à classe positiva;

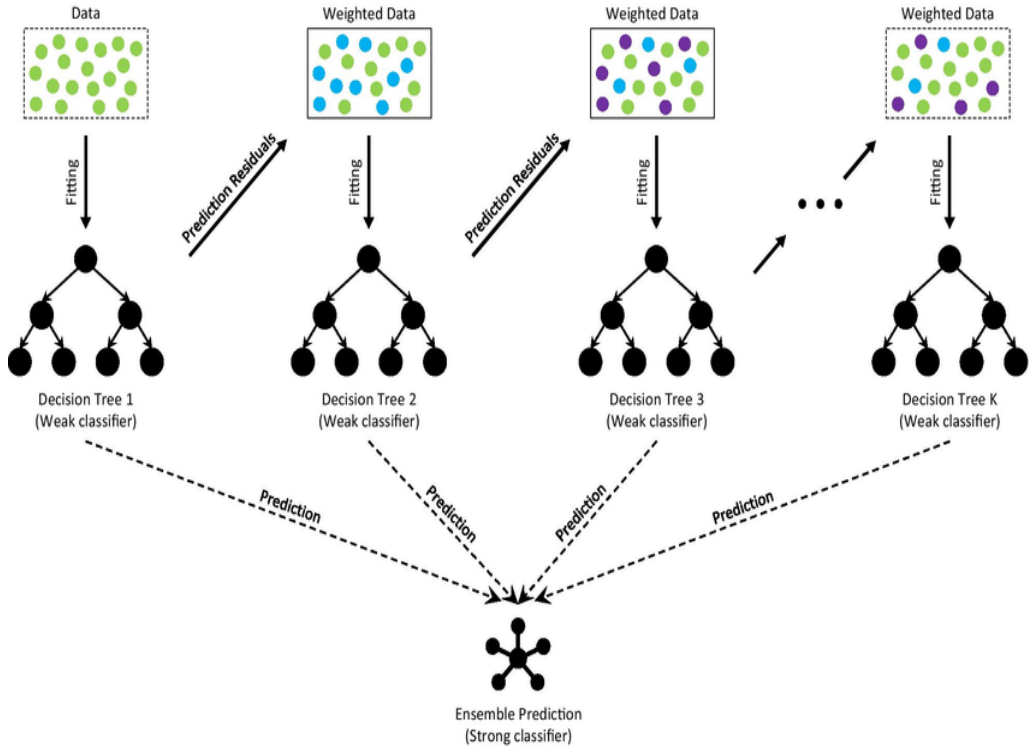
Treinamento: o objetivo do treinamento da Regressão Logística é ajustar os coeficientes do modelo para maximizar a verossimilhança dos dados observados. Isso é geralmente feito usando técnicas de otimização, como a Descida de Gradiente.

Gradient Boosting

- Conjunto de algoritmos de aprendizado de máquina que pertence à família dos algoritmos de *boosting*;

- Usado principalmente para problemas de regressão e classificação, e é conhecido por sua eficácia em criar modelos preditivos de alta qualidade;

- Constrói um modelo forte combinando a previsão de vários modelos de base (geralmente árvores de decisão simples) em um único modelo mais robusto.



Boosting: técnica em que múltiplos modelos mais fracos (geralmente chamados de "estimadores fracos") são combinados para formar um modelo forte. Cada modelo subsequente tenta corrigir os erros dos modelos anteriores;

Gradiente Descente: algoritmo utilizado para ajustar os pesos dos modelos de base de forma iterativa. Ele tenta minimizar a função de perda ao ajustar os pesos dos modelos para melhorar a previsão geral;

Árvores de Decisão como Estimadores Base: Em muitas implementações, as árvores de decisão são usadas como estimadores de base. Essas árvores são construídas de forma relativamente rasa e simples, geralmente com poucas divisões, para evitar *overfitting*;

Construção Iterativa: em cada etapa, um novo modelo de base é ajustado aos resíduos (erros) do modelo anterior. Essa abordagem permite que o modelo se concentre nas áreas em que os modelos anteriores tiveram dificuldade.

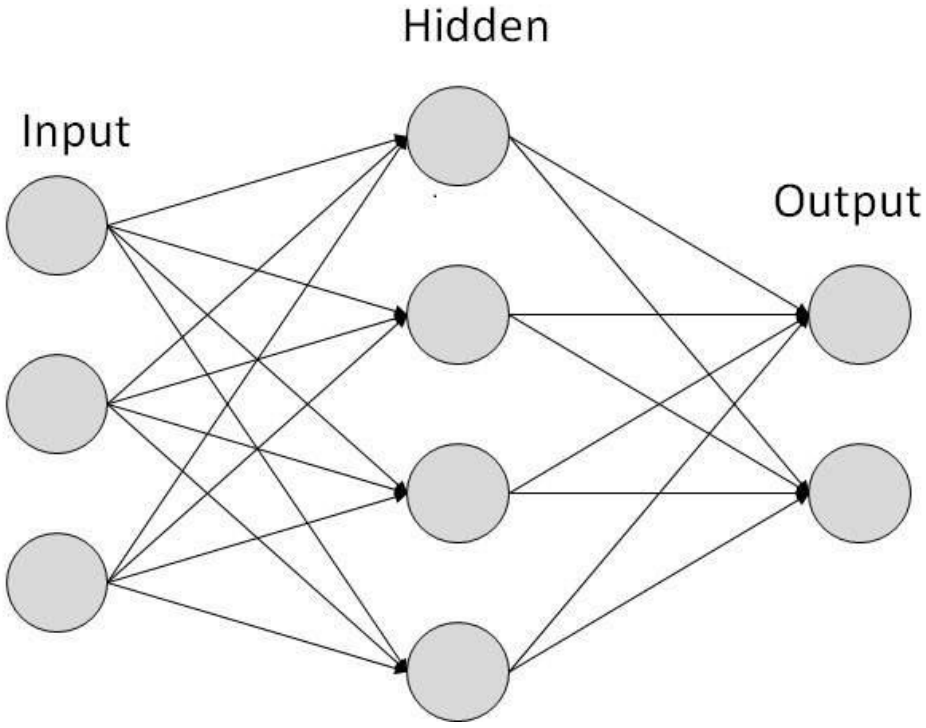
Aspecto	***XGBoost***	***CatBoost***	***LightGBM***
Regularização	Sim	Sim	Sim
Lidando com Dados Faltantes	Sim	Sim	Sim
Lidando com Categorias	Sim	Sim (tratamento automático)	Sim (tratamento automático)
Eficiência Computacional	Médio	Rápido	Rápido
Tratamento Outliers	Sensível	Menos sensível	Menos sensível
Integração GPU	Sim	Não	Sim
Performance	Bem Estabelecida	Potencialmente Elevada	Potencialmente Elevada
Balanceamento Classes	Não suporta nativamente	Suporta	Suporta

Redes Neurais Artificiais

- Modelos computacionais inspirados no funcionamento do cérebro humano;

- Compostas por unidades de processamento chamadas de neurônios artificiais, que são organizados em camadas e interconectados por conexões ponderadas;

- Amplamente utilizadas em aprendizado de máquina e têm sido muito eficazes em tarefas de reconhecimento de padrões, classificação, regressão e outras aplicações.



Arquitetura	**Descrição e Aplicação**
Perceptron Simples	Rede neural simples com um neurônio de saída.
Multi-Layer Perceptron (MLP)	Camadas ocultas para aprender funções não-lineares.
Redes Neurais Convolucionais	Processamento de dados espaciais, como imagens.
Redes Neurais Recorrentes	Processamento de sequências, como séries temporais.
Long Short-Term Memory (LSTM)	Lida com desaparecimento do gradiente em RNNs.
Gated Recurrent Units (GRUs)	Versão simplificada do LSTM.
Autoencoders	Aprendizado não supervisionado e redução de dimensionalidade.
Transformers	Processamento de linguagem natural e outras tarefas.
GPT (*Generative Pre-trained Transformer*)	Modelos de linguagem baseados em *Transformers*.

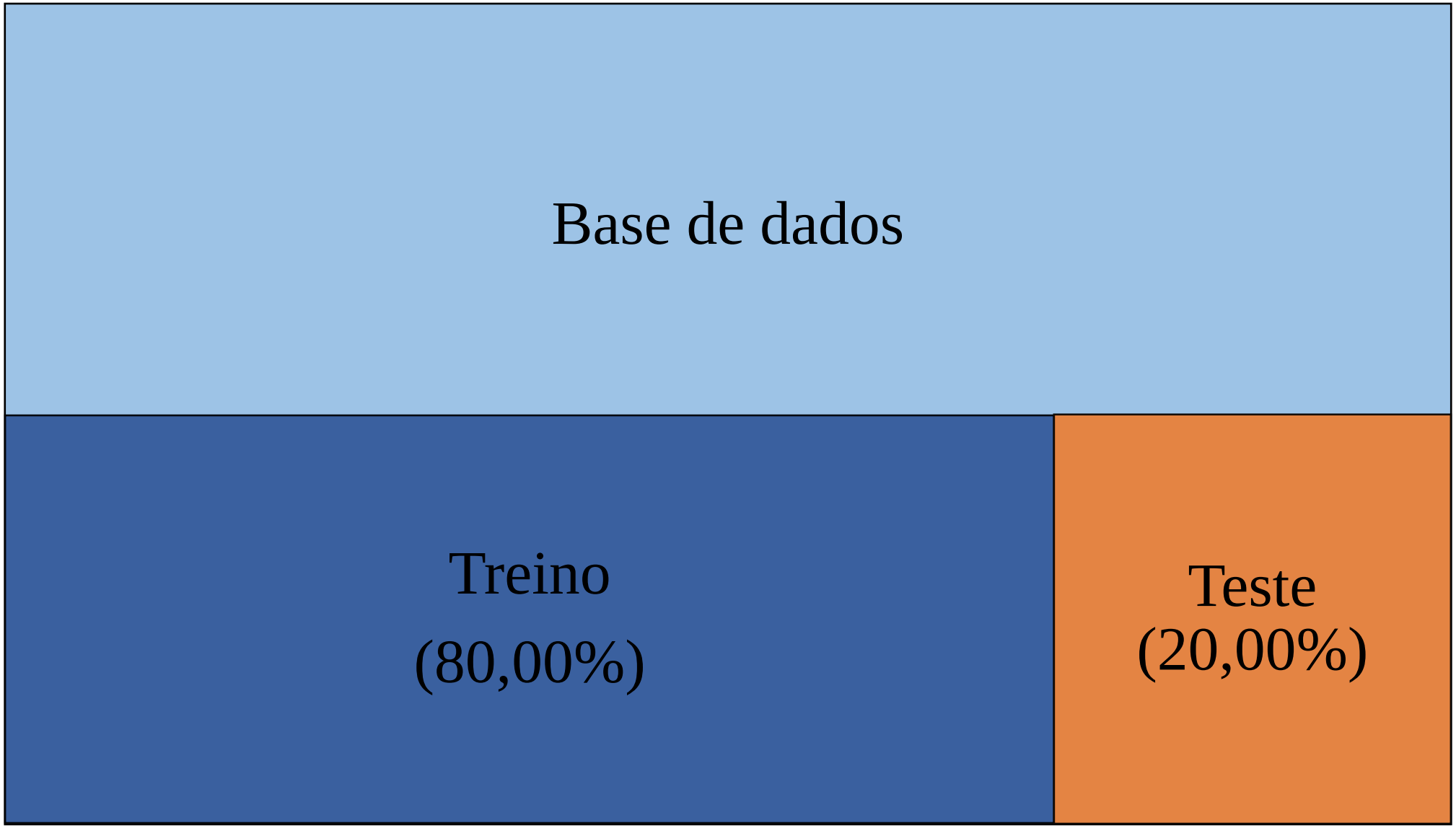
Estratégias de Treinamento

Divisão de Dados (*Train-Test Split*): o conjunto de dados é dividido em dois subconjuntos: um para treinamento e outro para teste. O modelo é treinado nos dados de treinamento e avaliado nos dados de teste para medir sua capacidade de generalização;

Validação Cruzada (*Cross-Validation*): envolve dividir o conjunto de dados em várias partes chamadas "dobras". O modelo é treinado em várias combinações dessas dobras, alternando entre treinamento e teste. Isso ajuda a obter uma avaliação mais robusta do desempenho do modelo.

Divisão de Dados: é mais simples de implementar e rápida, sendo adequada para conjuntos de dados grandes. No entanto, pode resultar em avaliações variáveis dependendo da divisão aleatória dos dados;

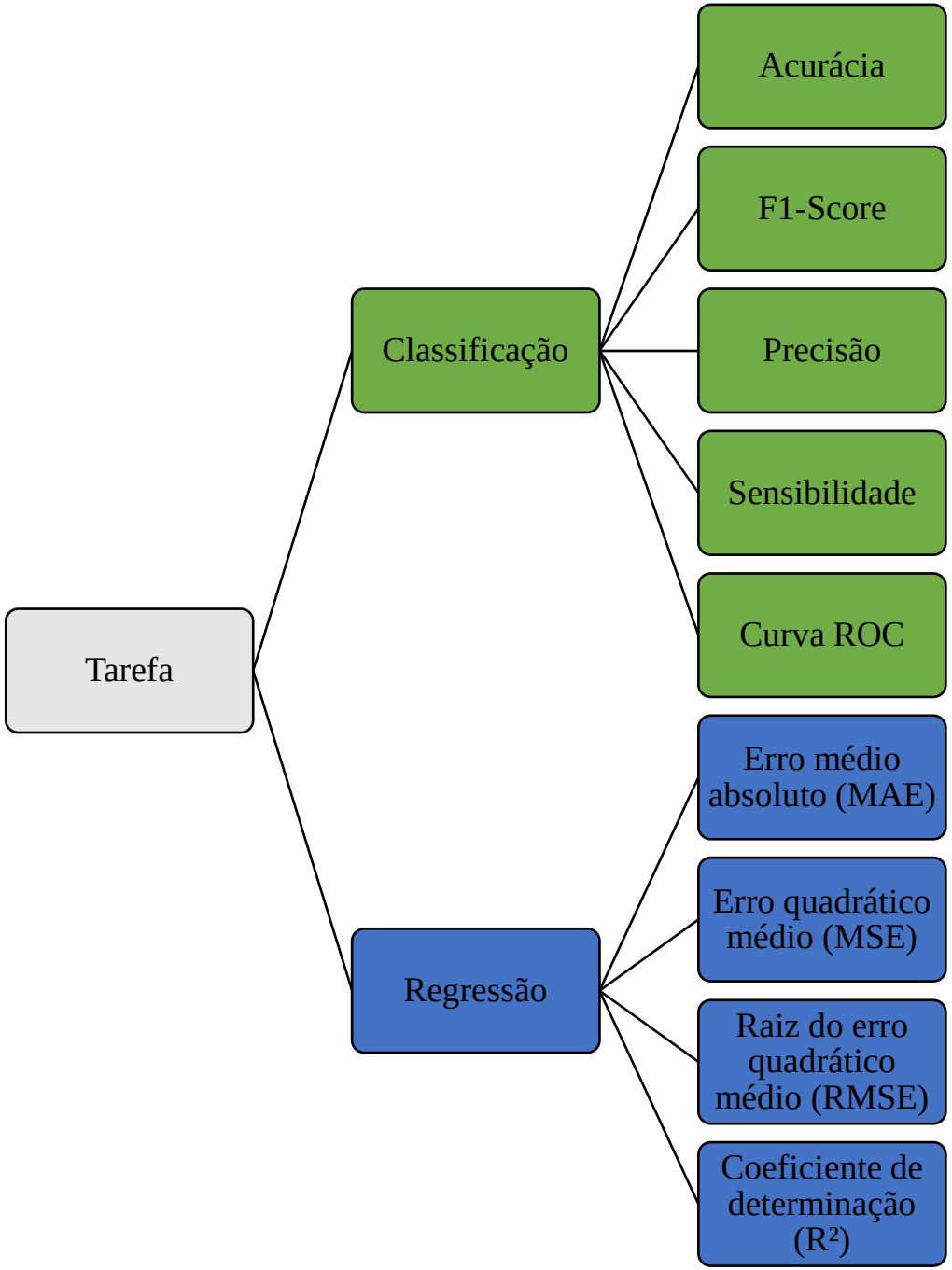
Validação Cruzada: é mais robusta, pois utiliza todas as amostras para treinamento e teste. Isso é especialmente importante em conjuntos de dados pequenos e ajuda a obter uma avaliação média mais precisa do desempenho do modelo. No entanto, a validação cruzada pode exigir mais processamento e ajuste, tornando-se mais complexa do que a divisão de dados simples.


Base de dados
Treino
(80,00%)
Teste
(20,00%)

1	**2**	**3**	**4**	**5**
1	**2**	**3**	**4**	**5**
1	**2**	**3**	**4**	**5**
1	**2**	**3**	**4**	**5**
1	**2**	**3**	**4**	**5**

Características	**Divisão de Dados**	**Validação Cruzada**
Objetivo	Avaliar o desempenho do modelo em dados não vistos	Estimar o desempenho médio do modelo em dados não vistos
Conjuntos de Dados	Dividido em Treinamento e Teste	Dividido em múltiplas dobras para treinamento e teste
Avaliação	Pode ter alta variabilidade devido à aleatoriedade da divisão	Geralmente resulta em uma avaliação mais estável e confiável
Uso de Dados	Pode desperdiçar uma parte significativa dos dados para teste	Usa todos os dados para treinamento e teste, reduzindo o desperdício
Adequado para	Conjuntos de dados grandes	Conjuntos de dados pequenos
Overfitting	Pode ocorrer, especialmente em conjuntos de dados pequenos	Menos propenso a *overfitting* devido à validação em múltiplas dobras
Implementação	Fácil de implementar	Requer mais processamento e ajuste
Métodos Variados	Não se aplica	*K-Fold*, *Leave-One-Out*, *Stratified Cross-Validation*, etc.

Métricas de Avaliação


Tarefa
Classificação
Acurácia
F1-Score
Precisão
Sensibilidade
Curva ROC
Regressão
Erro médio absoluto (MAE)
Erro quadrático médio (MSE)
Raiz do erro quadrático médio (RMSE)
Coeficiente de determinação (R²)

Métricas de Avaliação (Classificação)

1.Acurácia (*Accuracy*): Proporção de predições corretas em relação ao total de predições. É uma métrica geral, mas pode ser enganosa quando as classes estão desbalanceadas;

2.Precisão (*Precision*): A proporção de verdadeiros positivos (TP) em relação à soma de verdadeiros positivos e falsos positivos (FP). Mede a precisão das predições positivas;

3.Revocação (*Recall*): A proporção de verdadeiros positivos (TP) em relação à soma de verdadeiros positivos e falsos negativos (FN). Também conhecida como Sensibilidade ou Taxa de Verdadeiros Positivos, mede a capacidade do modelo em encontrar todos os exemplos positivos;

***4.F1-Score*:** A média harmônica entre precisão e recall. É uma métrica que equilibra precisão e revocação;

5. **Área sob a Curva ROC (AUC-ROC):** A área sob a curva da característica de operação do receptor (ROC). Mede a habilidade do modelo em distinguir entre classes positivas e negativas em diferentes níveis de limiar;

6. **Matriz de Confusão:** Tabela que mostra a contagem de verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN). A partir dela, diversas métricas podem ser calculadas;

7. **Taxa de Falsos Positivos (FPR):** Proporção de falsos positivos em relação à soma de verdadeiros negativos e falsos positivos;

8. **Taxa de Verdadeiros Negativos (TNR):** Proporção de verdadeiros negativos em relação à soma de verdadeiros negativos e falsos positivos.

Métrica	**Fórmula**
Acurácia	(TP + TN) / (TP + TN + FP + FN)
Precisão	TP / (TP + FP)
Revocação (Sensibilidade)	TP / (TP + FN)
F1-Score	2 * (Precisão * Revocação) / (Precisão + Revocação)
Área sob a Curva ROC	-
Área sob a Curva PR	-
Matriz de Confusão	-
Taxa de Falsos Positivos	FP / (TN + FP)
Taxa de Verdadeiros Negativos	TN / (TN + FP)

Métricas de Avaliação (Regressão)

1. **Erro Médio Absoluto (MAE)**: O MAE mede a média das diferenças absolutas entre os valores verdadeiros (y_true) e os valores previstos (y_pred);

2. **Erro Quadrado Médio (MSE)**: O MSE calcula a média das diferenças ao quadrado entre os valores verdadeiros (y_true) e os valores previstos (y_pred);

3. **Raiz do Erro Quadrado Médio (RMSE)**: O RMSE é a raiz quadrada do MSE e representa a média das diferenças ao quadrado entre os valores verdadeiros (y_true) e os valores previstos (y_pred);

4. **Erro Médio Percentual Absoluto (MAPE)**: O MAPE calcula a média das diferenças percentuais absolutas entre os valores verdadeiros (y_true) e os valores previstos (y_pred);

5. **Coeficiente de Determinação (R^2)**: O R^2 mede a proporção da variabilidade nos valores verdadeiros (y_true) que é explicada pelos valores previstos (y_pred). Um valor próximo a 1 indica um bom ajuste;

6. **Erro Quadrado Médio Logarítmico (MSLE)**: O MSLE calcula a média das diferenças ao quadrado entre os logaritmos dos valores verdadeiros (y_true) e dos valores previstos (y_pred);

7. **Mediana do Erro Absoluto (MedAE)**: O MedAE é a mediana das diferenças absolutas entre os valores verdadeiros (y_true) e os valores previstos (y_pred).

Métrica	Fórmula
Erro Médio Absoluto (MAE)	Σ(|y_true - y_pred|) / n
Erro Quadrado Médio (MSE)	Σ(y_true - y_pred)^2 / n
Raiz do MSE (RMSE)	√(Σ(y_true - y_pred)^2 / n)
Erro Médio Percentual Absoluto (MAPE)	Σ(|(y_true - y_pred) / y_true|) / n * 100
Coeficiente de Determinação (R^2)	1 - Σ(y_true - y_pred)^2 / Σ(y_true - y_true_médio)^2
Erro Quadrado Médio Logarítmico (MSLE)	Σ(log(y_true + 1) - log(y_pred + 1))^2 / n
Mediana do Erro Absoluto (MedAE)	Mediana das diferenças absolutas