Algoritmos comuns de ML

- Existe uma grande diversidade de algoritmos de Machine Learning que abrangem os mais diversos problemas e objetivos. Entre os mais comuns podemos citar:

1) Regressão linear

- É um dos algoritmos mais simples e comuns de Machine Learning. Ele trabalha com conceitos matemáticos e estatísticos para prever um valor, por isso é comumente utilizado em problemas de Regressão com Aprendizado Supervisionado.

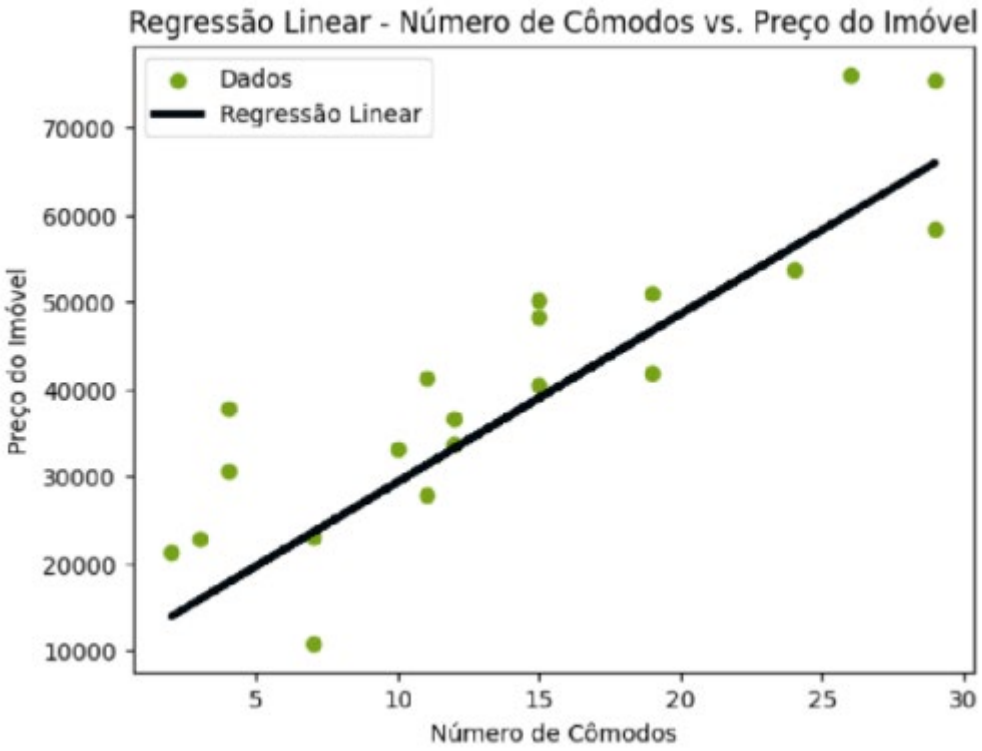
- A principal estratégia desse modelo é identificar a relação matemática e linear entre um valor de saída e os valores de entrada. A regressão linear encontra a reta que melhor representa a relação entre variáveis.

- A representação que melhor demonstra essa relação será uma reta que minimiza as distâncias entre ela e todos os pontos de intersecção entre os valores das variáveis.

- Por exemplo, se estivermos prevendo preços de casas, a regressão linear aprenderia a relação entre características das casas, como o número de cômodos e seus preços, encontrando a melhor reta para fazer previsões.

- Infelizmente, o que torna a Regressão Linear um algoritmo simples, também o torna muito limitado, pois para que o modelo funcione de maneira adequada, é preciso existir uma relação linear entre as variáveis de entrada e a variável de saída rotulada.

- Em contraposição a essa desvantagem, na Regressão Linear é possível observar a representação gráfica da relação linear entre as variáveis.



2) Regressão Logística

- Mesmo existindo “Regressão” em seu nome e podendo ser aplicada em problemas de Regressão, a Regressão Logística é comumente utilizada em problemas de Classificação binária.

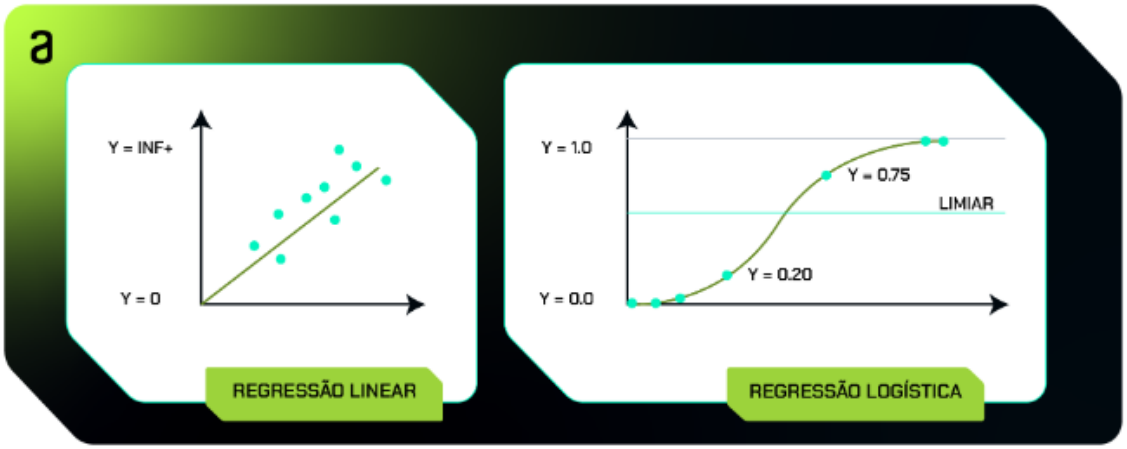
- Isso é alcançado porque a Regressão Logística fornece valores probabilísticos entre 0 e 1, que podem representar a probabilidade de uma amostra pertencer a uma classe específica.

- A palavra “Logística” no nome do algoritmo, refere-se a função logística que é aplicada a combinação linear das características do conjunto de dados.

- Ao contrário da Regressão linear que tem uma reta que pode atingir diversos valores, a função logística tem forma de “S” e faz a transformação dos valores em uma escala de 0 a 1.

- Para decidir a probabilidade de uma amostra pertencer a uma categoria, o modelo utiliza do Limiar que determina a fronteira entre as classes.

- Geralmente esse valor é especificado como 0.5 e tem grande impacto nas decisões: se a probabilidade Y estiver acima do Limiar, a instância é atribuída à classe positiva; caso contrário, à classe negativa.



3) Árvores de decisão

- A Árvore de Decisão, empregada em aprendizado supervisionado para classificação e regressão, é uma estrutura hierárquica composta por nós de decisão e nós folha.

- Durante o treinamento, o algoritmo inicia com um nó raiz representando todo o conjunto de dados, dividindo-o recursivamente com base em variáveis de entrada até que cada subconjunto contenha apenas uma classe ou valor de saída.

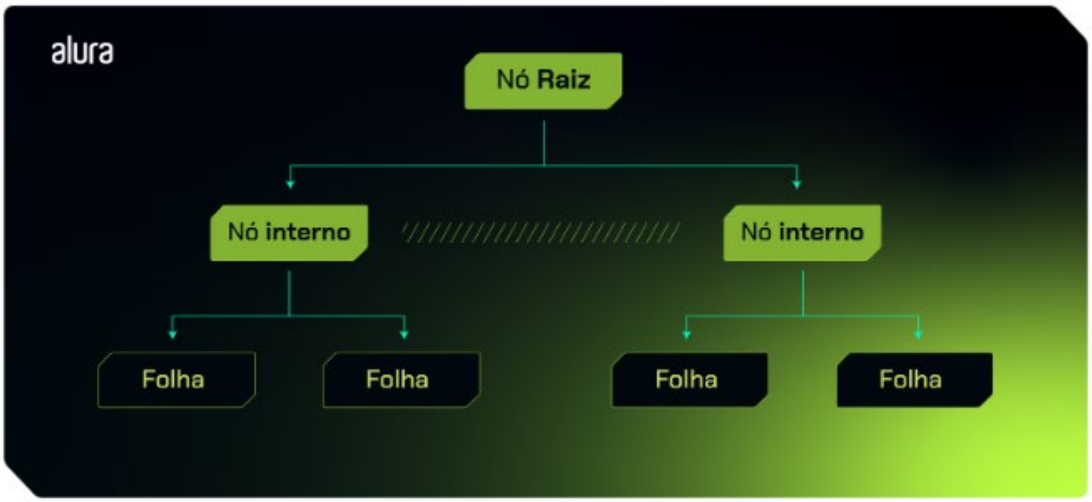
- A estrutura final mostra uma raiz conectada a nós de decisão, representando escolhas com base nas variáveis de entrada, e a nós folha, indicando resultados finais.

- A estrutura de uma Árvore de Decisão assemelha-se a uma árvore comum, começando pela raiz da árvore que é o nó superior e representa todo o conjunto de dados.

- A partir disso, a árvore se ramifica em nós de decisão, que representam as escolhas que podem ser feitas com base nas variáveis de entrada.

- Cada nó de decisão tem ramos que representam as diferentes opções que podem ser escolhidas. Esses ramos levam a outros nós de decisão ou a nós folha, que representam os resultados finais.

- Os nós folha são os nós finais da árvore e representam os resultados finais da análise. Eles não têm ramos saindo deles. Cada nó folha representa uma classe, em caso de problemas de classificação ou valor de saída, em problemas de regressão.



4) K-means clustering

- É uma técnica de Aprendizado Não Supervisionado que agrupa um conjunto de dados não rotulados em diferentes clusters.

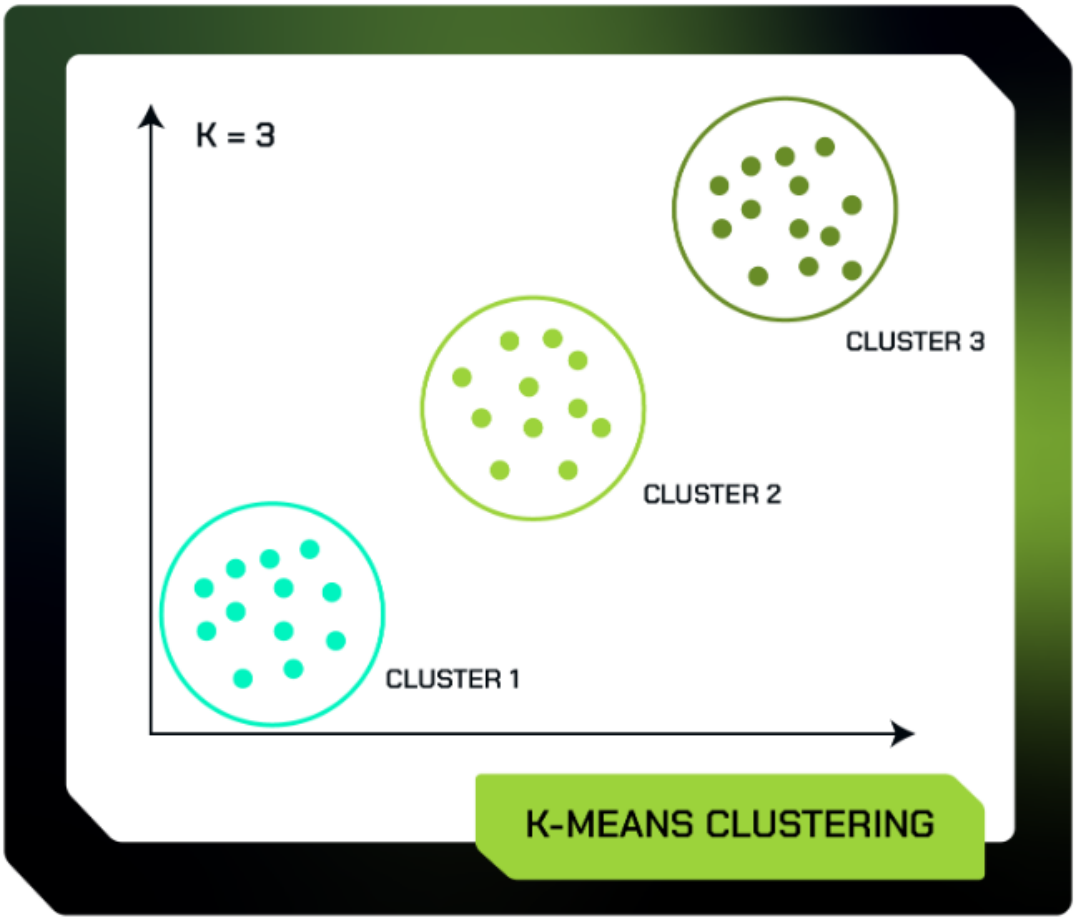
- Nesse método, o valor de K (número de clusters desejado) é pré-definido pela pessoa especialista, indicando quantos grupos serão criados durante o processo.

- O algoritmo funciona da seguinte maneira:

* Primeiro, o usuário especifica o número de clusters desejado (K), para explicar vamos considerar K = 3.

* Em seguida, o algoritmo seleciona aleatoriamente 3 pontos (K pontos) do conjunto de dados como centros iniciais dos clusters.

* A cada iteração, cada ponto de dados é atribuído ao cluster cujo centro está mais próximo. Os centros dos clusters são então recalculados com base nos pontos atribuídos a eles, e o processo é repetido até que a atribuição dos pontos aos clusters não mude significativamente entre iterações. Desse Modo Vamos ter ao final 3 clusters:



- Ao final, esse algoritmo divide o conjunto de dados em 3 clusters (K clusters) de modo que cada conjunto pertença a apenas um grupo com propriedades semelhantes.

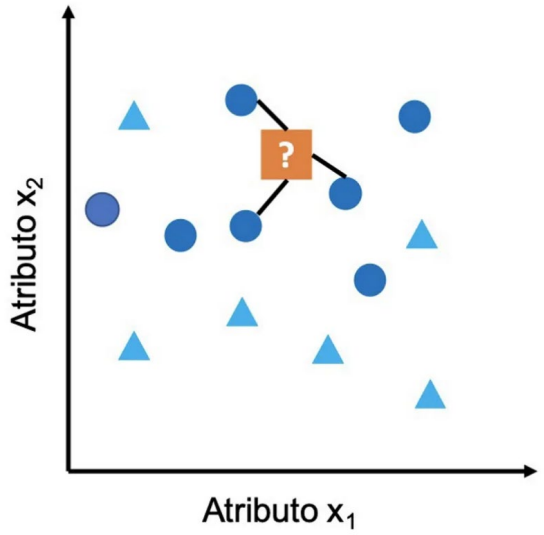
- A ideia principal é minimizar a soma das distâncias entre os pontos de dados e os centroides correspondentes.

5) K-Nearest Neighbours

- O algoritmo KNN é um algoritmo de simples entendimento e que funciona muito bem na prática, podendo ser utilizado tanto para problemas de Classificação quanto para de Regressão

- Sua ideia principal é considerar que os exemplos vizinhos são similares ao exemplo cuja informação se deseja inferir, uma ideia parecida com “Diga-me com quem andas e eu te direi quem tu és!”.

- O KNN considera que os registros do conjunto de dados correspondem a pontos no espaço Rn, em que cada atributo corresponde a uma dimensão deste espaço. Exemplo em R2:



- No KNN, o conjunto de dados de treinamento é armazenado e, quando um novo exemplo chega, ele é comparado a todos os exemplos armazenados para identificar os K (que é um parâmetro de entrada do algoritmo) vizinhos mais próximos (mais semelhantes) de acordo com alguma métrica de distância (por exemplo, distância euclidiana).

- No caso de ser um problema de classificação, a classe do novo registro é determinada por inspeção das classes dos *k* vizinhos mais próximos, de acordo com a métrica escolhida. No caso de um problema de regressão, em vez da classe, examina-se o valor de Y dos *k* vizinhos. Na maioria das implementações do KNN, os atributos são normalizados no início do algoritmo, para que contribuam igualmente na predição da classe ou do valor.

- As etapas a seguir resumem o algoritmo KNN:

* Definição da métrica de distância utilizada e valor de *k*.

* Cálculo da distância do novo exemplo a cada um dos exemplos existentes no conjunto inicial de entrada.

* Identificação dos *k* exemplos do conjunto de referência que apresentaram menor distância em relação ao novo exemplo (mais similares).

* Apuração da classe mais frequente entre os *k* exemplos identificados no passo anterior, usando votação majoritária (para problemas de classificação) ou estimação do valor Y como a média aritmética dos *k*-vizinhos mais próximos.

6) Redes Bayesianas

- Para entender efetivamente é necessário entender o Teorema de Bayes e a Teoria de grafos

* O Teorema de Bayes descreve a probabilidade de um evento dado que outro evento já ocorreu, o que é chamado de probabilidade condicional.

* A teoria de grafos estuda grafos, que podem ser definidos como um conjunto de objetos que estão relacionados de certa forma. Esses objetos correspondem à vértices (também chamados de nós ou pontos) e as relações correspondem à arestas (também chamadas de link ou linha).

- Definição:

Representação das relações entre as probabilidades de ocorrência de eventos utilizando probabilidades a priori e a posteriori relacionadas através de um grafo, onde os vértices são as variáveis aleatórias e as arestas são as relações de dependência.

- Tendo como definição formal: par ordenado (S, P), no qual:

* S é a estrutura da rede (nós e arestas)

* P é o conjunto de distribuições de probabilidade p (xi | pa(xi)), em que pa(xi) são os nós pais de xi.

- É possível obter a distribuição de probabilidade conjunta da rede multiplicando as probabilidades condicionais em cada nó em todos os caminhos da rede.

- Utilidade – Inferência

Inicialmente ela pode ser tida como uma forma de representar um conhecimento especialista sobre incertezas do objeto de estudo e de inferir sobre o que era desconhecido, ou seja, onde as probabilidades não são representadas diretamente no modelo.

- Redes Bayesianas como classificadores:

* Além da inferência, a rede pode ser interpretada como um classificador, onde a classe e as características são representadas no modelo, sendo que a classe possui uma dependência em relação às características (não obrigatoriamente, à todas elas). Dessa forma é realizada uma classificação com base nas probabilidades a posteriori.

- Pontos fortes:

* Funciona mesmo quando entradas de dados estão faltando.

* Pode aprender relacionamentos casuais

* É uma representação ideal para combinar conhecimento a priori e conhecimento proveniente dos dados

* São eficientes para construir modelos de domínios com incerteza inerente

- Problemas:

* Se o número de configurações é grande, o erro de estimação será grande

* Mesmo com uma amostra grande, o que evitaria o problema anterior, seria custoso estimar os parâmetros

7) Florestas aleatórias

- O Random Forest é um algoritmo de aprendizado de máquina supervisionado que se destaca pela diversidade.

- Ao treinar, em vez de usar todos os dados disponíveis, ele seleciona aleatoriamente amostras do conjunto de treino.

- Da mesma forma, ao construir cada árvore, o algoritmo não considera todas as variáveis, mas escolhe, de forma aleatória, algumas delas para determinar os melhores nós.

- Esse processo adiciona uma camada de aleatoriedade e diversidade ao modelo, ajudando a evitar um foco excessivo em um conjunto específico de variáveis e, consequentemente, o overfitting.

- A seleção aleatória de variáveis e amostras pode parecer estranha à primeira vista, mas é altamente eficaz. Essa estratégia resulta em diversas árvores, cada uma com pequenas imperfeições aleatórias, o que evita o overfitting a um conjunto particular de dados.

- A pluralidade de árvores contribui para a robustez do modelo, e no final, as árvores combinam suas previsões para gerar um resultado mais confiável, seja tirando a média das previsões em problemas de regressão ou selecionando o resultado mais comum em problemas de classificação.