Data Science & Business Analytics

Machine Learning Models

David Issá
davidribeiro.issa@gmail.com

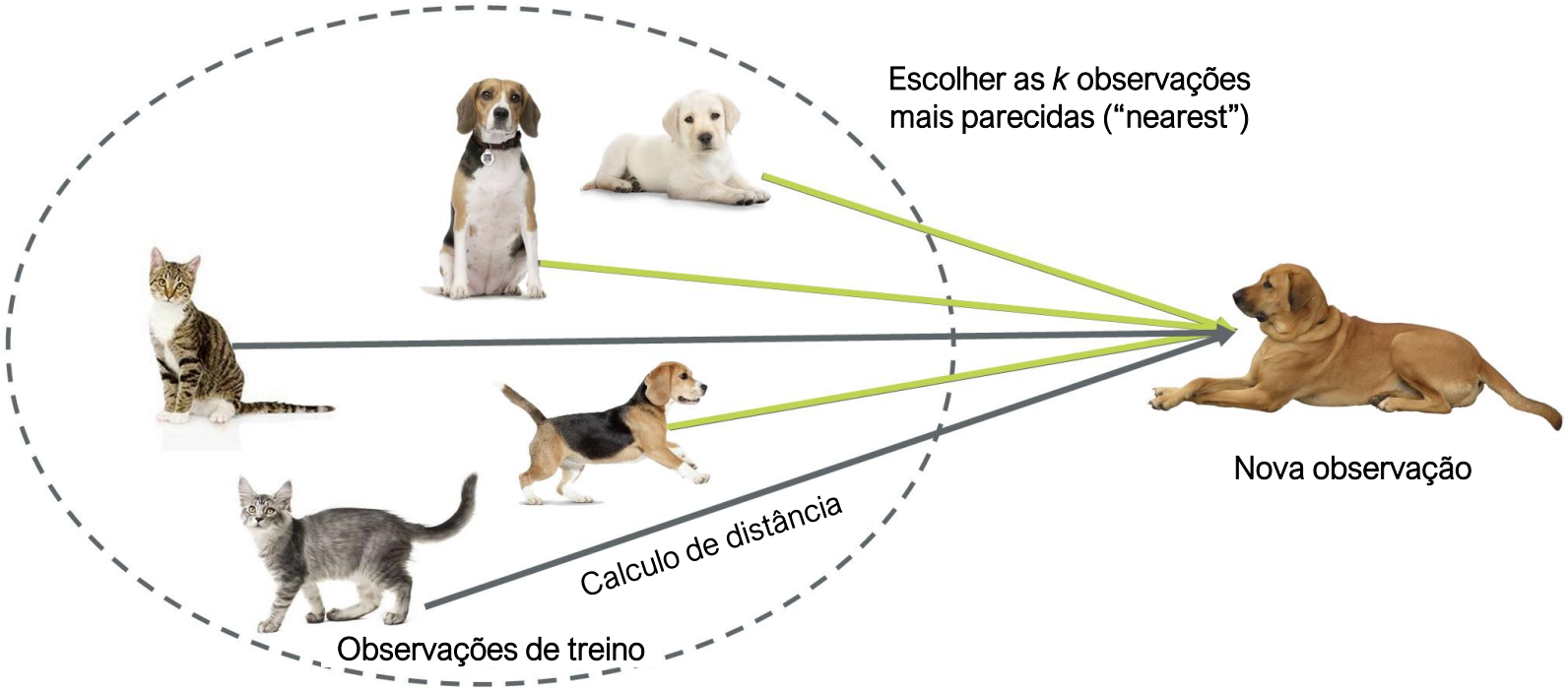
1. Algoritmos “Instance-Based”

1. Algoritmos “Instance-Based”

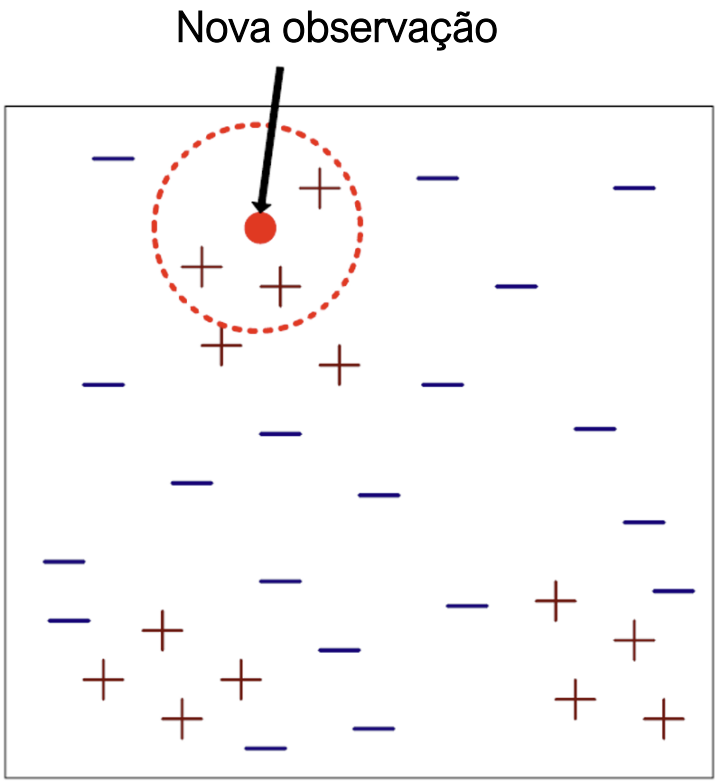
- Corresponde a uma família de algoritmos que, em vez de efetuar uma generalização, comparam as novas instâncias do problema com as instâncias vistas no treino;
- Uma nova instância procura nas instâncias de treino aquela com que mais se assemelha;
- As próprias instâncias representam o conhecimento;
- É um conjunto de métodos de “lazy learning” (vs. “eager learning”).
- Um dos principais modelos é o Nearest Neighbours.

1. Algoritmos “Instance-Based” – Nearest Neighbor

- Se anda como um cão, ladra como um cão, então é provavelmente um cão:



1. Algoritmos "Instance-Based" – Nearest Neighbor



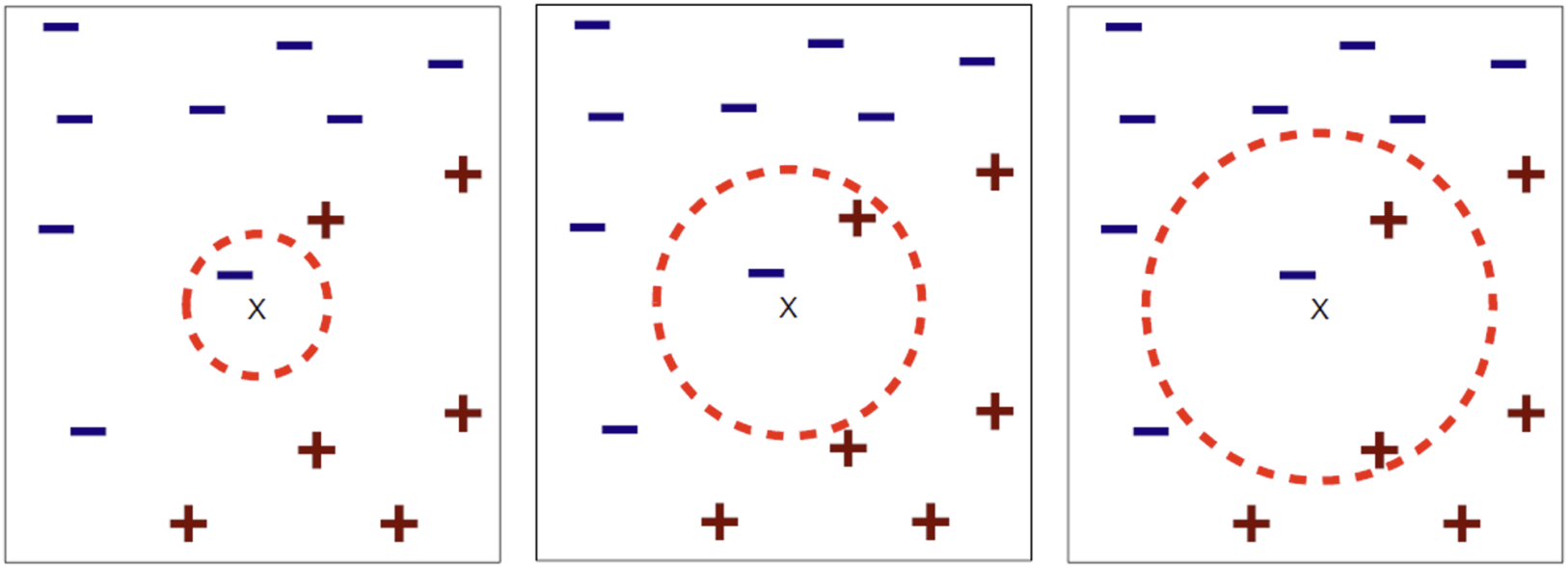
<u>São necessárias 3 coisas:</u>

1. O conjunto de observações de treino;
2. A métrica de distância para calcular a distância entre observações;
3. O valor de *k*, o número de vizinhos mais próximos a comparar.

<u>Para classificar uma observação nova:</u>

1. Calcular a distância aos registos de treino;
2. Identificar os *k*-vizinhos mais próximos;
3. Utilizar as classe dos vizinhos mais próximos para determinar a classe da nova observação (votação por maioria).

1. Algoritmos “Instance-Based” – Nearest Neighbor



k = 1 *k* = 2 *k* = 3

Nota: no Sklearn, se verificar que dois vizinhos têm distâncias idênticas mas classes diferentes, os resultados dependerão da ordem dos dos dados de treino.

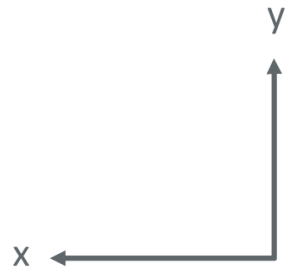
1. Algoritmos "Instance-Based" – Nearest Neighbor

- Algumas das formas de calcular distâncias entre observações, tal como visto anteriormente:

Euclidean distance

$$d(x,y) = \sqrt{\sum_{k=1}^{p}(x_k - y_k)^2}$$

Manhattan distance



$$d(x,y) = \sum_{k=1}^{p}|x_k - y_k|$$

Nota: se todas as variáveis independentes forem categóricas, pode ser usada a distância Hamming.

1. Algoritmos “Instance-Based” – Nearest Neighbor

Tal como nos algoritmos de clustering, quando usadas métricas de distância, as variáveis devem ser normalizadas (fazer “scaling”, por exemplo, com min-max ou z-score).

Isto serve para evitar que as medidas de distância sejam dominadas por uma das variáveis.

Por exemplo:

- a altura de uma pessoa pode variar de 1,5m a 2m
- o peso de uma pessoa pode variar entre 50 kg e 90 kg
- o rendimento de uma pessoa pode variar entre 5 mil euros e 50 mil euros

1. Algoritmos "Instance-Based" – Nearest Neighbor

Os dados de treino:

ID	var1	var2	var3	var4	Target
1	3	7	4	3	No
2	5	4	3	5	No
3	4	5	4	3	No
4	8	3	2	7	Yes
5	5	6	5	8	Yes
6	5	2	5	3	No
7	3	4	8	6	Yes
8	6	5	9	4	No

A nova observação:

ID	var1	var2	var3	var4	Target
100	3	6	4	5	?

ID	Distância Total (Manhattan)	Target
1	3	No
2	5	No
3	4	No
4	12	Yes
5	6	Yes
6	9	No
7	7	Yes
8	10	No

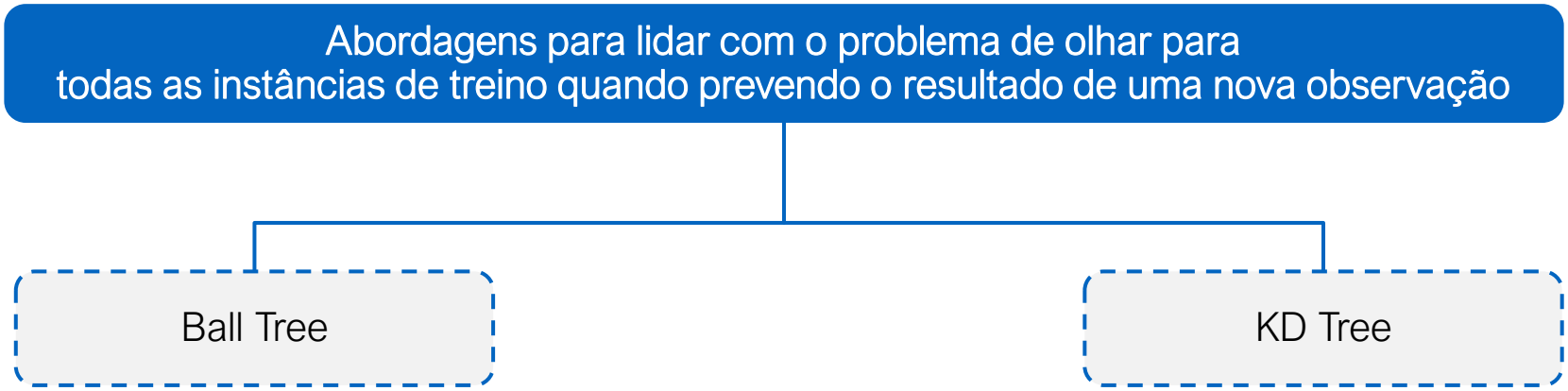
k = 1 → 1 "No" → Target = "No"
k = 2 → 2 "No" → Target = "No"
k = 3 → 3 "No" → Target = "No"
k = 4 → 3 "No" & 1 "Yes" → Target = "No"

1. Algoritmos “Instance-Based” – Nearest Neighbor

No entanto, algoritmos como o Nearest Neighbors apresentam algumas dificuldades, tais como:

- Ter de calcular a distância do caso de teste a todos as istâncias de treino;
- Necessita de muita memória para armazenar o conjunto de treino;
- Exige muito tempo na fase de classificação;
- Muito sensível aos outliers;
- Muito sensível à função de distância escolhida.

1. Algoritmos “Instance-Based” – Nearest Neighbor



1. Algoritmos “Instance-Based” – Nearest Neighbor

1.1 Algoritmo Ball Tree

- Acelera a descoberta de pontos vizinhos mais semelhantes;
- O algoritmo Ball Tree é um método de indexação espacial - organiza e estrutura as observações considerando o espaço dimensional em que os pontos estão localizados;
- Tem uma estrutura hierárquica.

1. Algoritmos “Instance-Based” – Nearest Neighbor

1.1 Algoritmo Ball Tree

1º Passo:

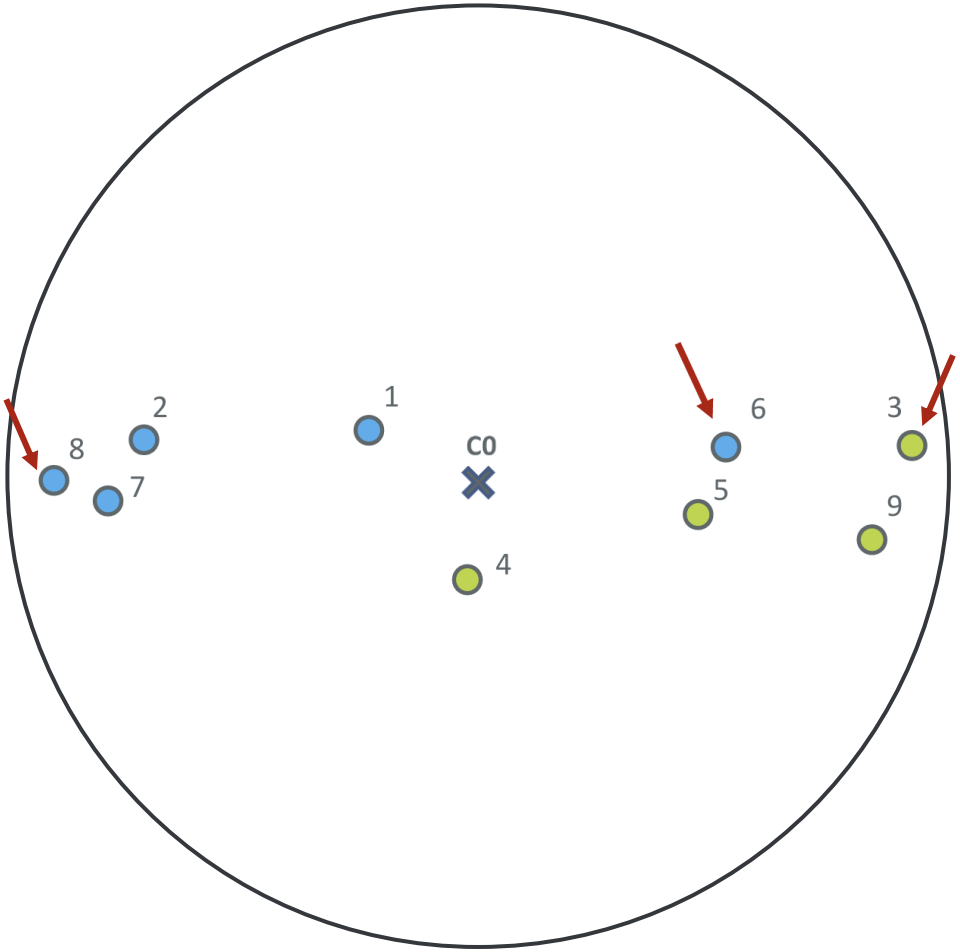
Selecionar uma observação de forma aleatória (ID6).

1. Algoritmos “Instance-Based” – Nearest Neighbor

1.1 Algoritmo Ball Tree

2º Passo:

- Medir a distância entre a observação selecionada (ID6) e todas as as outras, obtendo a que está mais longe (ID8).
- Utilizando a última observação selecionada (ID8), medir a distância a todos as outras e obter o que está mais distante (ID3).
- Estes são os limites do nosso primeiro cluster, e o centroide é definido (C0).

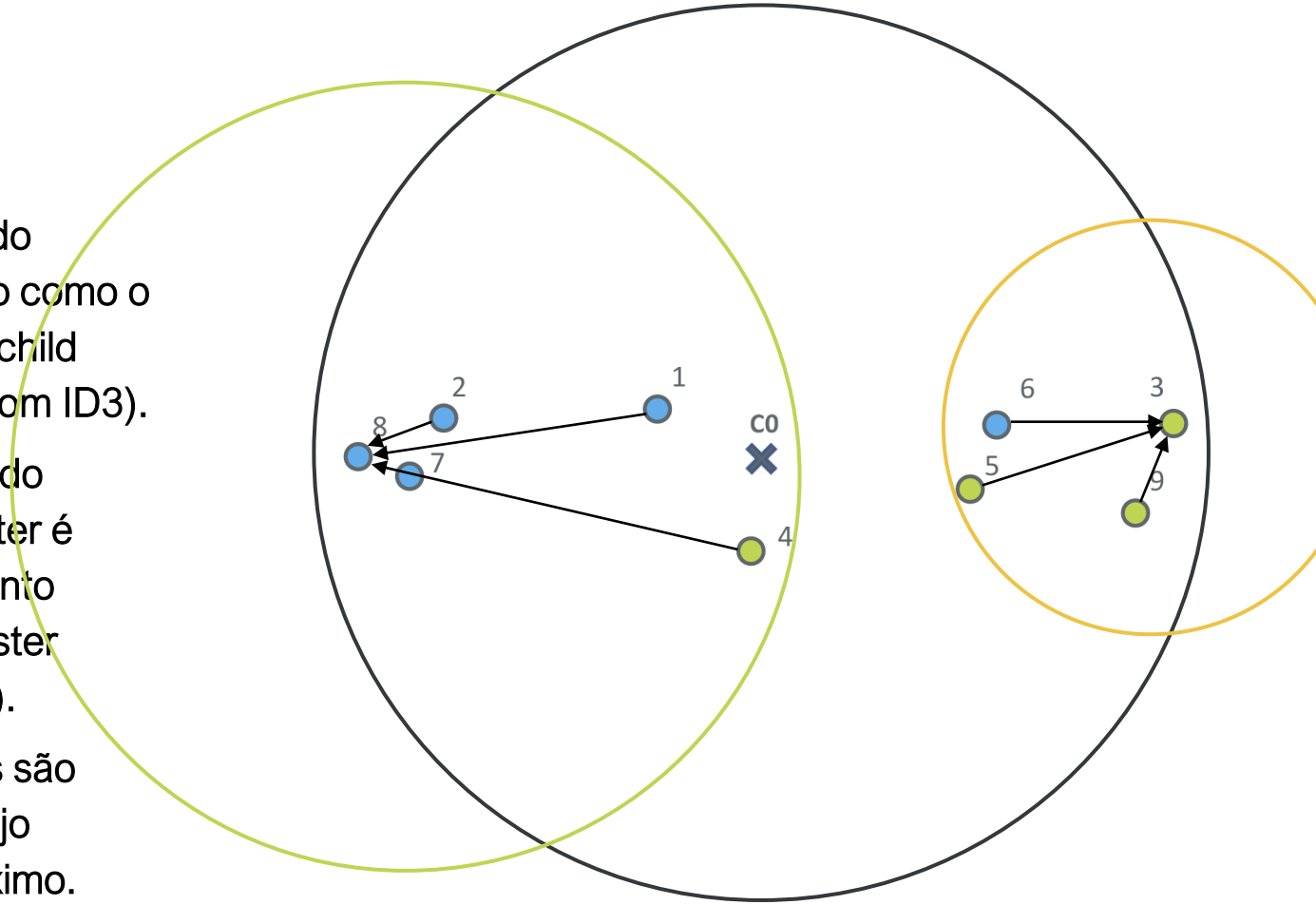


1. Algoritmos “Instance-Based” – Nearest Neighbor

1.1 Algoritmo Ball Tree

3º Passo:

- O ponto mais distante do centroide é selecionado como o centro do 1º cluster e “child node” (cluster laranja com ID3).
- O ponto mais afastado do centro do primeiro cluster é selecionado como o ponto central do segundo cluster (cluster verde com ID8).
- Todos os outros pontos são atribuídos ao cluster cujo centroide é o mais próximo.

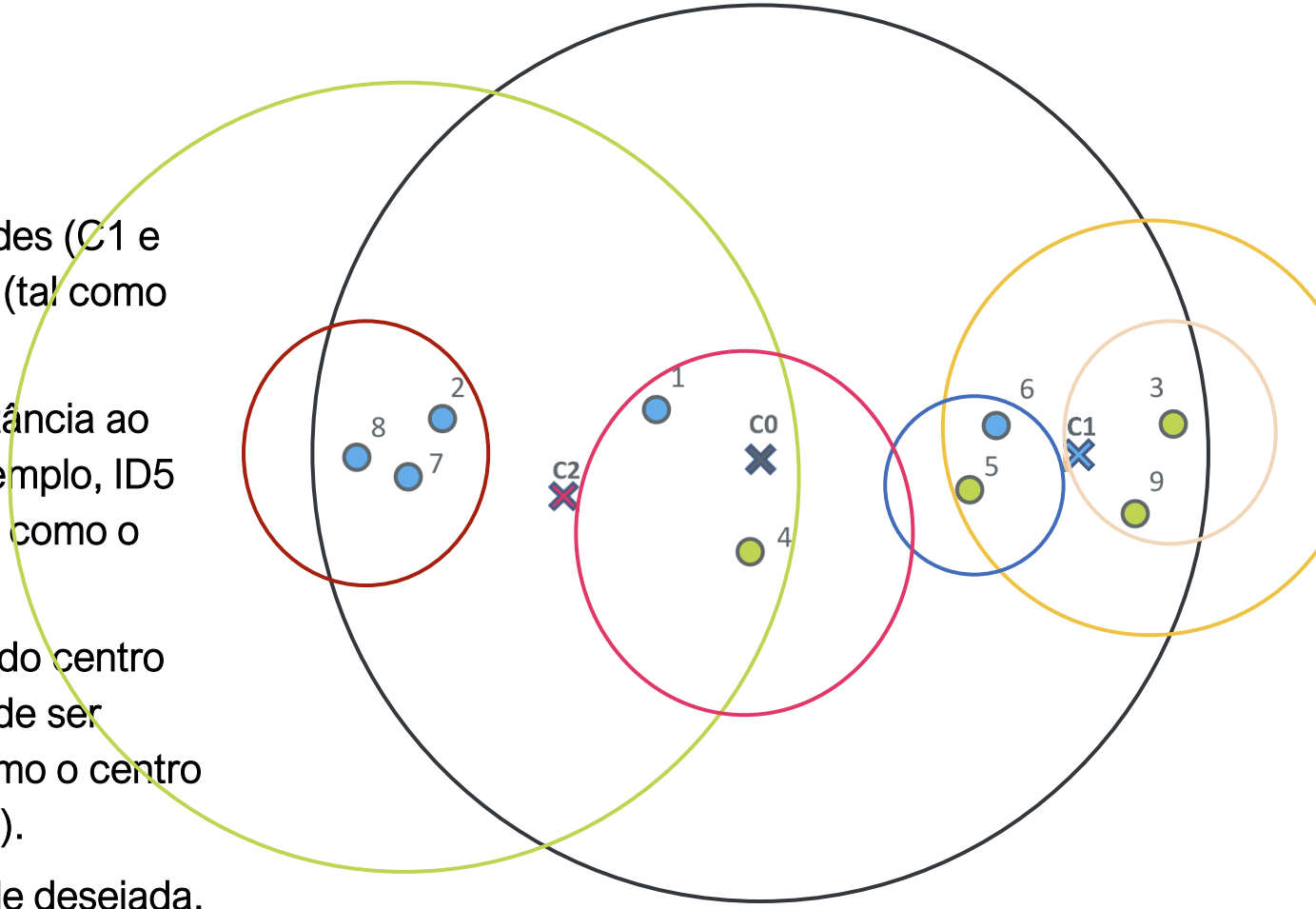


1. Algoritmos “Instance-Based” – Nearest Neighbor

1.1 Algoritmo Ball Tree

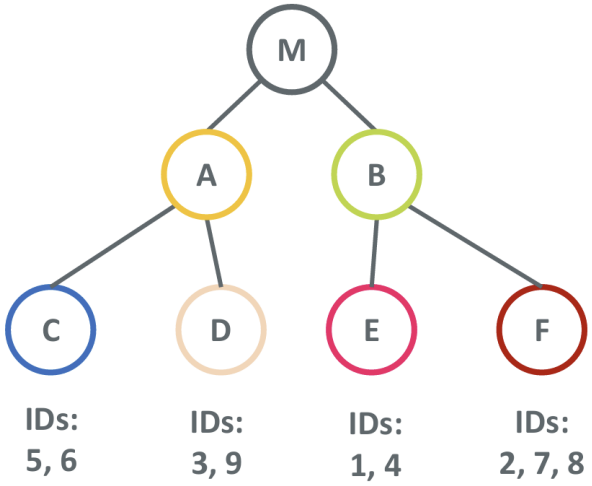
4º Passo:

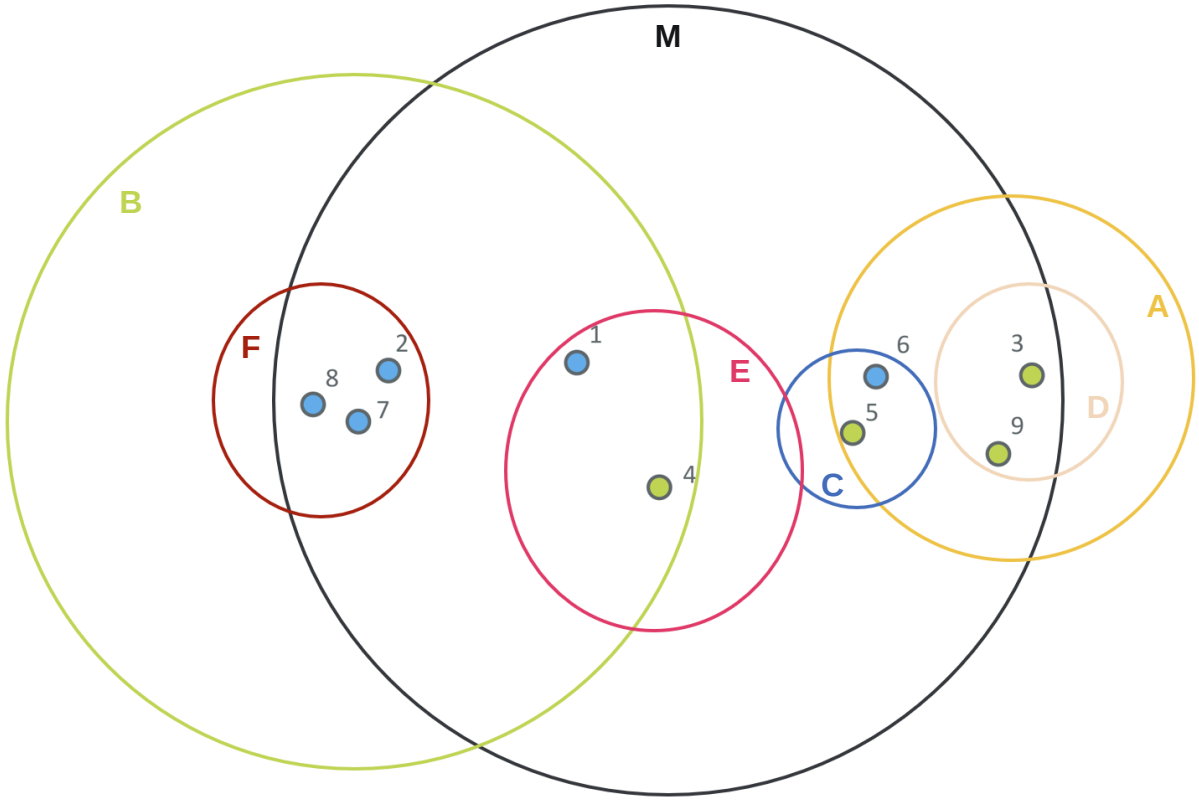
- Obter os novos centróides (C1 e C2) nos novos clusters (tal como no 2º passo).
- O ponto com maior distância ao novo centróide (por exemplo, ID5 para C1) é selecionado como o centro do novo cluster.
- O ponto mais afastado do centro do cluster que acabou de ser definido é escolhido como o centro do próximo cluster (ID3).
- Repetir até profundidade desejada.



1. Algoritmos “Instance-Based” – Nearest Neighbor

1.1 Algoritmo Ball Tree





1. Algoritmos “Instance-Based” – Nearest Neighbor

1.1 Algoritmo Ball Tree

- Utilizando este algoritmo, o processo de classificação através modelo Nearest Neighbor é bastante acelerado.

- Para um determinado ponto não classificado, será primeiro determinado o respetivo cluster. Assim, apenas será necessário calcular a distância para os pontos pertencentes a esse cluster.

1. Algoritmos “Instance-Based” – Nearest Neighbor

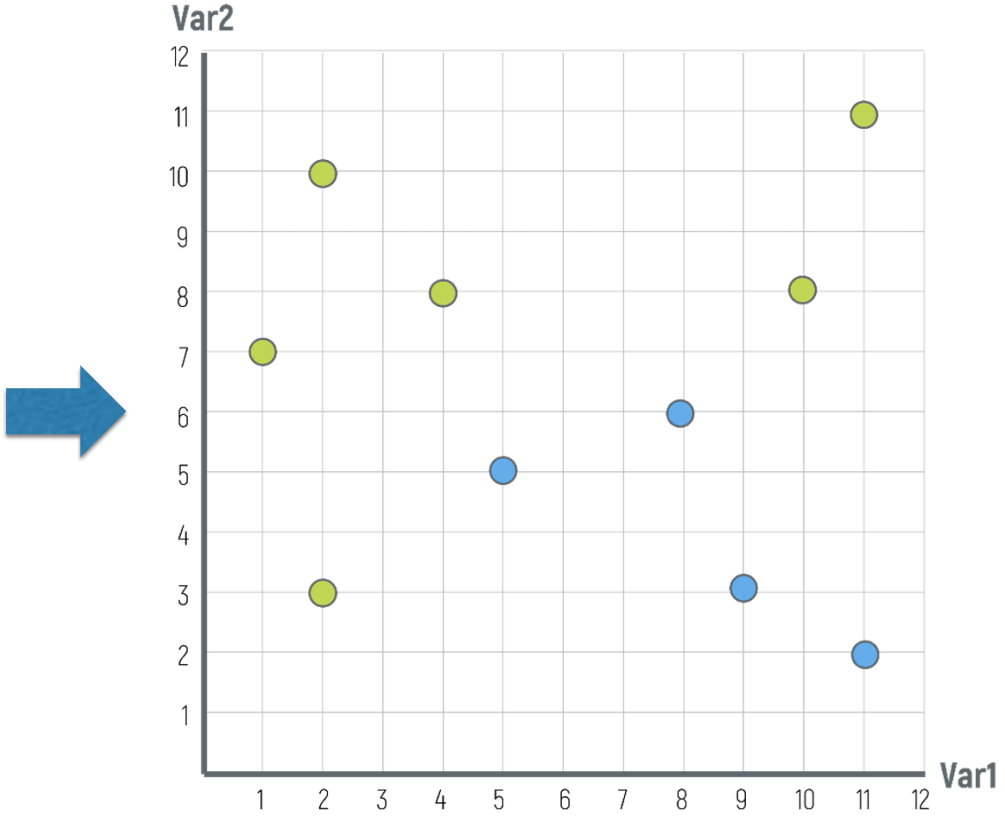
1.2 Algoritmo KDTree

- Acelera a descoberta de pontos vizinhos mais semelhantes;
- Constrói uma estrutura de dados hierárquica ordenada denominada Árvore k-Dimensional (KDTree);
- É uma árvore binária;
- A ideia geral das KDTrees é particionar o espaço dimensional composto pelas variáveis.

1. Algoritmos “Instance-Based” – Nearest Neighbor

1.2 Algoritmo KDTree

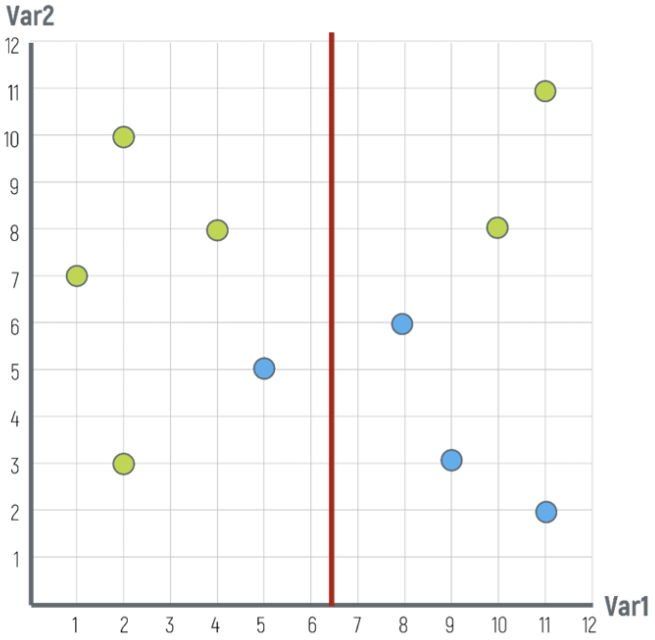
ID	Var1	Var2	Target
1	4	8	Green
2	9	3	Blue
3	11	11	Blue
4	11	2	Blue
5	1	7	Green
6	2	3	Green
7	5	5	Blue
8	2	10	Green
9	8	6	Blue
10	10	8	Green



1. Algoritmos “Instance-Based” – Nearest Neighbor

1.2 Algoritmo KDTree

1º Passo: Começar com a primeira variável da lista, encontrar a sua mediana e dividir os dados em conformidade (se uma observação tiver o valor da mediana, é incluído na partição >=)



Var1
4
9
11
11
1
2
5
2
8
10

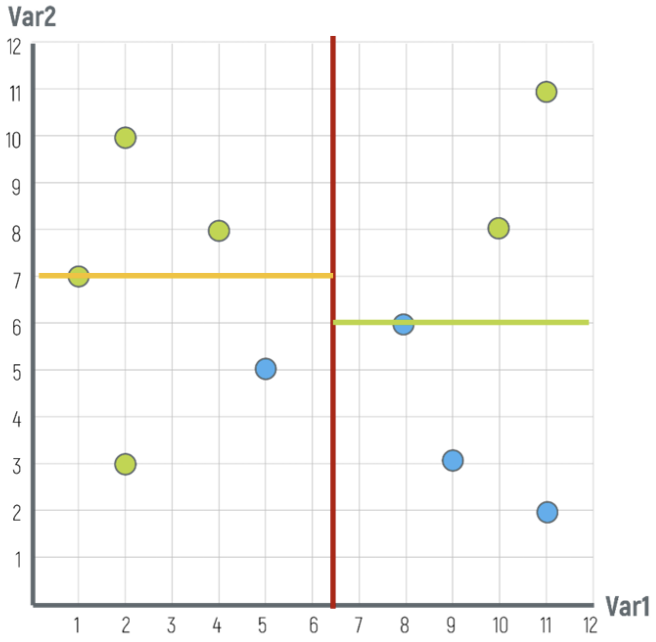
Var1
1
2
2
4
5
8
9
10
11
11

Mediana = 6.5

1. Algoritmos “Instance-Based” – Nearest Neighbor

1.2 Algoritmo KDTree

2º Passo: Passar para a segunda variável, encontrar a mediana e dividir em conformidade. Efectue estas divisões (alterando em cada nível a variável utilizada) até obter a profundidade desejada.



Var2
8
7
3
5
10

Var2
3
5
7
8
10

Mediana = 7

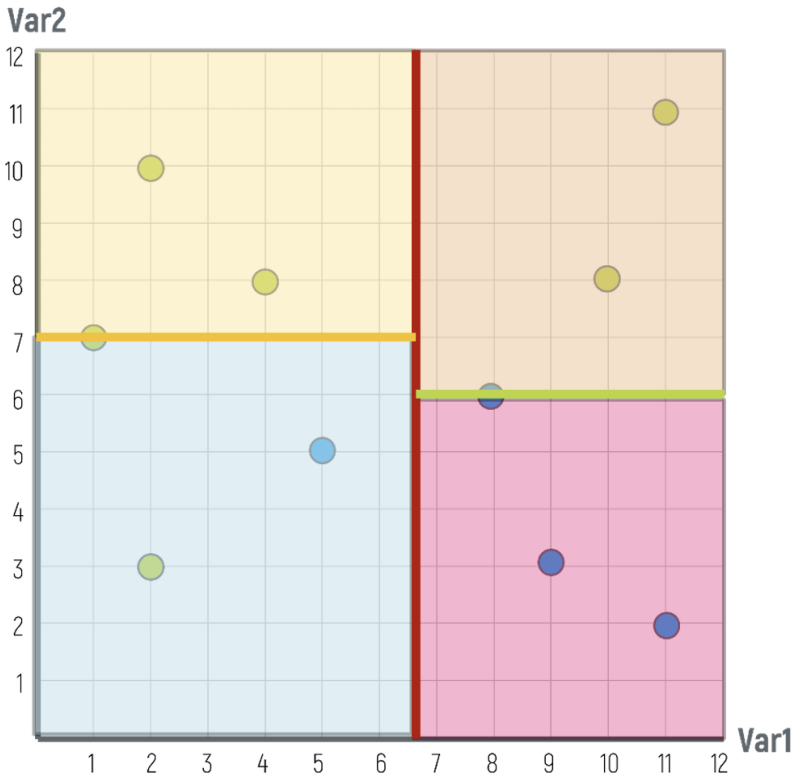
Var2
3
11
2
6
8

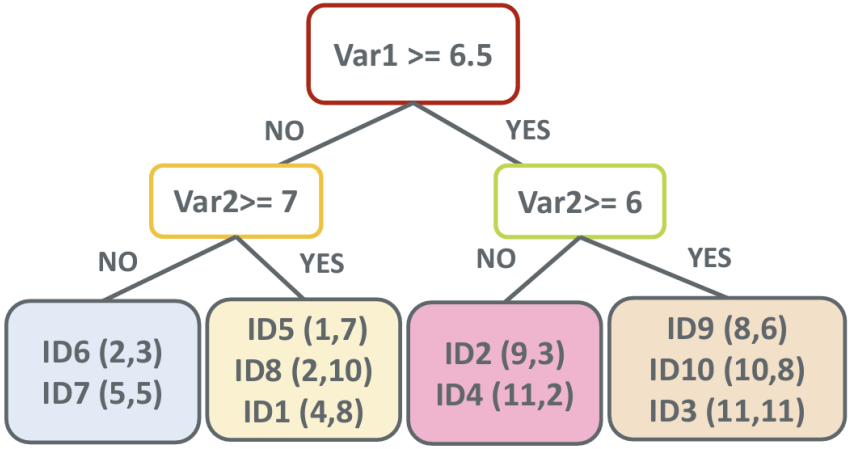
Var2
2
3
6
8
11

Mediana = 6

1. Algoritmos “Instance-Based” – Nearest Neighbor

1.2 Algoritmo KDTree





1. Algoritmos “Instance-Based” – Nearest Neighbor

1.2 Algoritmo KDTree

- As KDTrees dividem o espaço de dimensional de modo a podermos excluir partições inteiras que estão mais distantes do que os nossos k vizinhos mais próximos;

- Normalmente a melhor solução para dados de baixa dimensão.

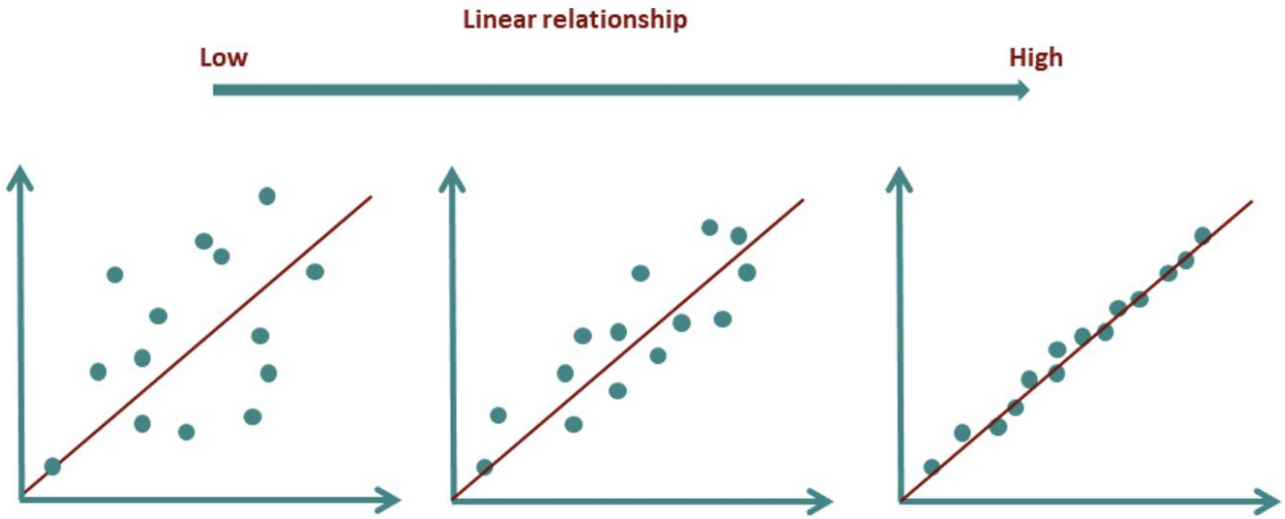
2. Algoritmos “Model-Based”

2. Algoritmos “Model-Based”

- Corresponde a uma família de algoritmos que envolvem a criação de um modelo matemático que pode prever resultados com base nos dados de treino;
- O modelo pode ser considerado como um conjunto de regras que a máquina utiliza para fazer previsões;
- Os dados de treino são utilizados para criar um modelo que pode ser generalizado a novos dados (“eager learning”).;
- Alguns exempos de algoritmos são: Regressão Linear, a Regressão Logística, as Decision Trees e as Neural Networds.

2.1 Algoritmos “Model-Based” – Regressão Linear

- É um algoritmo que estabelece uma relação linear entre as variáveis independentes e a variável alvo para prever o resultado de eventos futuros.
- É utilizado normalmente quando a variável alvo é numérica, por exemplo, vendas, salário, idade, preço do produto, preço de imóvel etc.

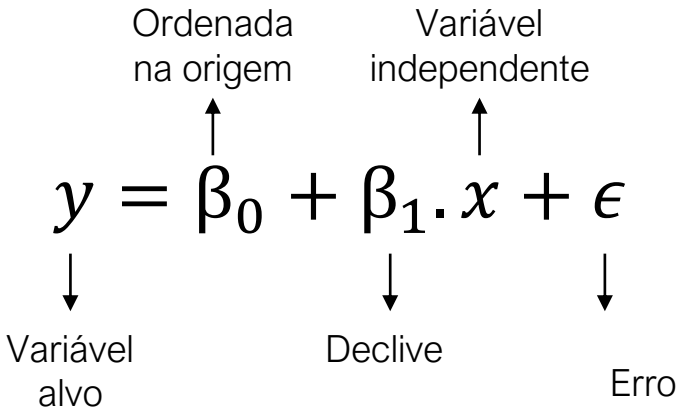


2.1 Algoritmos “Model-Based” – Regressão Linear

- No caso de existir apenas 1 variável independente, a tarefa da Regressão Linear seria a de determinar a linha reta que melhor descreve a relação linear entre a variável alvo e a variável independente.

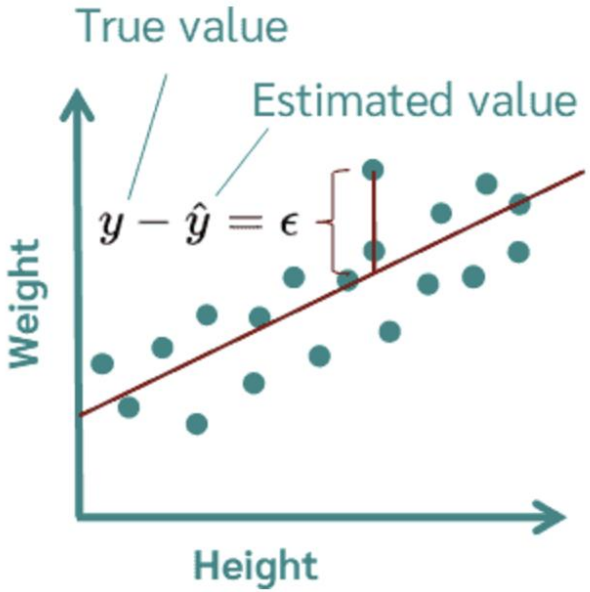
2.1 Algoritmos "Model-Based" – Regressão Linear

- No caso de existir apenas 1 variável independente, a tarefa da Regressão Linear seria a de determinar a linha reta que melhor descreve a relação linear entre a variável alvo e a variável independente.



2.1 Algoritmos "Model-Based" – Regressão Linear

- No caso de existir apenas 1 variável independente, a tarefa da Regressão Linear seria a de determinar a linha reta que melhor descreve a relação linear entre a variável alvo e a variável independente.



- O objetivo é estimar os parâmetros β_0 e β_1, através da minimização da Sum of Squared Residuals: $\sum \epsilon^2 = \sum (y - \hat{y})^2$.
- Assim, obtemos a regra geral em que obtemos o valor estimado da variável alvo $\hat{y}$ para qualquer valor da variável independente x:

$$\hat{y} = \beta_0 + \beta_1 . x$$

2.1 Algoritmos “Model-Based” – Regressão Linear

Se pretendemos minimizar $\sum \epsilon^2 = \sum (y - \hat{y})^2$, e sabemos que $\hat{y} = \beta_0 + \beta_1 . x$, então o objetivo é minimizar $\sum \epsilon_i^2 = \sum (y_i - \beta_0 - \beta_1 . x_i)^2$.

2.1 Algoritmos "Model-Based" – Regressão Linear

Se pretendemos minimizar $\sum \epsilon^2 = \sum (y - \hat{y})^2$, e sabemos que $\hat{y} = \beta_0 + \beta_1 . x$, então o objetivo é minimizar $\sum \epsilon_i^2 = \sum (y_i - \beta_0 - \beta_1 . x_i)^2$.

Começamos por calcular as derivadas parciais com respetio a β_0 e β_1:

- $\frac{d\epsilon^2}{d\beta_0} = 0 \leftrightarrow -2 \sum (y_i - \beta_0 - \beta_1 . x_i) = 0 \leftrightarrow \sum y_i = n\beta_0 + n\beta_1 . \sum x_i$
- $\frac{d\epsilon^2}{d\beta_1} = 0 \leftrightarrow -2 \sum x_i (y_i - \beta_0 - \beta_1 . x_i) = 0 \leftrightarrow \sum x_i y_i = \beta_0 . \sum x_i + \beta_1 . \sum x_i$

2.1 Algoritmos "Model-Based" – Regressão Linear

Se pretendemos minimizar $\sum \epsilon^2 = \sum (y - \hat{y})^2$, e sabemos que $\hat{y} = \beta_0 + \beta_1 . x$, então o objetivo é minimizar $\sum \epsilon_i^2 = \sum (y_i - \beta_0 - \beta_1 . x_i)^2$.

Começamos por calcular as derivadas parciais com respetio a β_0 e β_1:

- $\frac{d\epsilon^2}{d\beta_0} = 0 \leftrightarrow -2 \sum (y_i - \beta_0 - \beta_1 . x_i) = 0 \leftrightarrow \sum y_i = n\beta_0 + n\beta_1 . \sum x_i$
- $\frac{d\epsilon^2}{d\beta_1} = 0 \leftrightarrow -2 \sum x_i (y_i - \beta_0 - \beta_1 . x_i) = 0 \leftrightarrow \sum x_i y_i = \beta_0 . \sum x_i + \beta_1 . \sum x_i$

Resolvendo um sistema de 2 equações com 2 icógnitas:

- $\sum y_i = n\beta_0 + n\beta_1 . \sum x_i \leftrightarrow \beta_0 = \frac{\sum y_i}{n} - \beta_1 . \frac{\sum x_i}{n} \leftrightarrow \beta_0 = \bar{y} - \beta_1 . \bar{x}$

2.1 Algoritmos "Model-Based" – Regressão Linear

Se pretendemos minimizar $\sum \epsilon^2 = \sum (y - \hat{y})^2$, e sabemos que $\hat{y} = \beta_0 + \beta_1 . x$, então o objetivo é minimizar $\sum \epsilon_i^2 = \sum (y_i - \beta_0 - \beta_1 . x_i)^2$.

Começamos por calcular as derivadas parciais com respetio a β_0 e β_1:

- $\frac{d\epsilon^2}{d\beta_0} = 0 \leftrightarrow -2 \sum (y_i - \beta_0 - \beta_1 . x_i) = 0 \leftrightarrow \sum y_i = n\beta_0 + n\beta_1 . \sum x_i$
- $\frac{d\epsilon^2}{d\beta_1} = 0 \leftrightarrow -2 \sum x_i (y_i - \beta_0 - \beta_1 . x_i) = 0 \leftrightarrow \sum x_i y_i = \beta_0 . \sum x_i + \beta_1 . \sum x_i$

Resolvendo um sistema de 2 equações com 2 icógnitas:

- $\sum y_i = n\beta_0 + n\beta_1 . \sum x_i \leftrightarrow \beta_0 = \frac{\sum y_i}{n} - \beta_1 . \frac{\sum x_i}{n} \leftrightarrow \beta_0 = \bar{y} - \beta_1 . \bar{x}$
- $\sum x_i y_i = \beta_0 . \sum x_i + \beta_1 . \sum x_i \leftrightarrow \beta_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} = \frac{Cov(x,y)}{Var(x)}$

2.1 Algoritmos "Model-Based" – Regressão Linear

- No caso de existirem mais do que 1 variável independentes (cenário mais realístico), o processo é o mesmo, mas estimamos tantos parâmetros quanto o número de variáveis independentes (mais a ordenada na origem).

$$y = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2 + \cdots + \beta_n . x_n + \epsilon$$

2.1 Algoritmos "Model-Based" – Regressão Linear

- No caso de existirem mais do que 1 variável independentes (cenário mais realístico), o processo é o mesmo, mas estimamos tantos parâmetros quanto o número de variáveis independentes (mais a ordenada na origem).

$$y = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2 + \cdots + \beta_n . x_n + \epsilon$$

- Os coeficientes são interpretados da seguinte forma: se uma variável independente mudar uma unidade, o coeficiente associado indica em quanto a variável alvo muda.

2.1 Algoritmos "Model-Based" – Regressão Linear

- No caso de existirem mais do que 1 variável independentes (cenário mais realístico), o processo é o mesmo, mas estimamos tantos parâmetros quanto o número de variáveis independentes (mais a ordenada na origem).

$$y = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2 + \cdots + \beta_n . x_n + \epsilon$$

- Os coeficientes são interpretados da seguinte forma: se uma variável independente mudar uma unidade, o coeficiente associado indica em quanto a variável alvo muda.

- Depois de estimar os parâmetros (com os dados de treino), apenas necessitamos de guardar esses parâmetros, e podemos estimar o valor da variável alvo para quaisquer valores das variáveis indepentendes, apenas multiplicando o parâmetro pelo valor da sua variável correspondente.

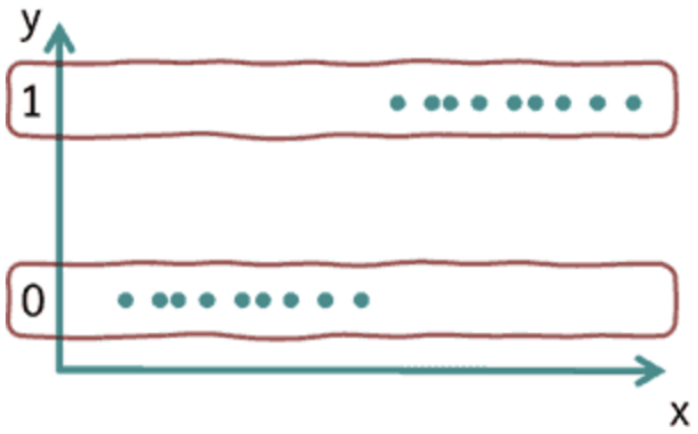
2.2 Algoritmos “Model-Based” – Regressão Logística

- Na Regressão Linear, as variáveis independentes (por exemplo, idade e sexo) são utilizadas para estimar o valor de uma variável dependente numérica (por exemplo, peso corporal).

- Na Regressão Logística, por outro lado, a variável dependente é uma variável categórica e binária (0 ou 1) e a probabilidade de ocorrência do evento em questão acontecer é estimada.

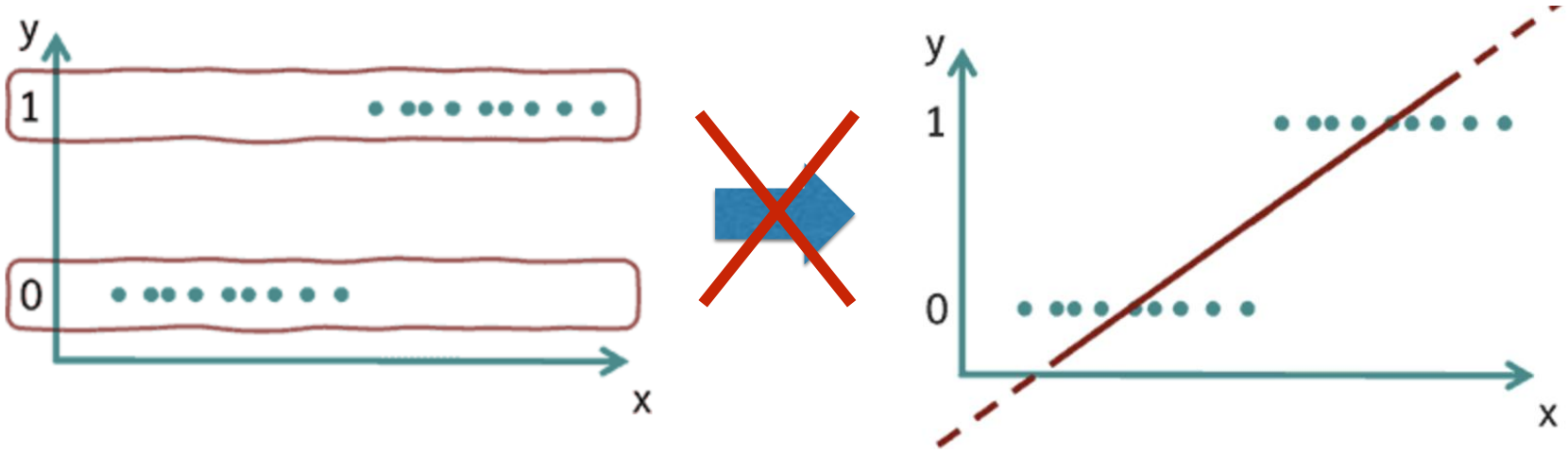
2.2 Algoritmos "Model-Based" – Regressão Logística

- Assim, entendemos que não podemos proceder à estimação da Regressão Logística da mesma forma que procedemos com a Regressão Linear.
- No caso de haver apenas 1 variável independente:



2.2 Algoritmos "Model-Based" – Regressão Logística

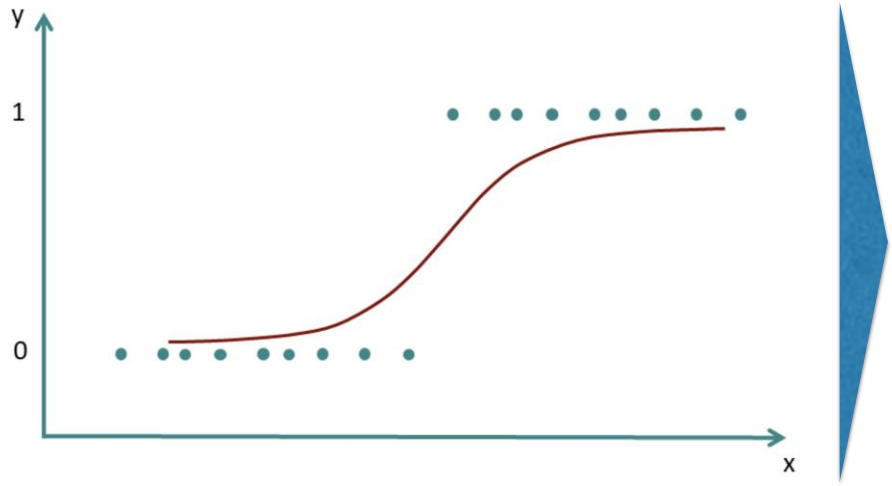
- Assim, entendemos que não podemos proceder à estimação da Regressão Logística da mesma forma que procedemos com a Regressão Linear.
- No caso de haver apenas 1 variável independente:



- A relação entre as variáveis independentes e a variável alvo não é linear!

2.2 Algoritmos "Model-Based" – Regressão Logística

- Para estabelecer esta nova relação, necessitamos de recorrer à função Sigmoid:



- Função Sigmoid no contexto da Regressão Logística:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 . x_1)}}$$

- Independentemente do intervalo de valores de x, y estará entre 0 e 1.

- A Regressão Logística não tem um termo de erro devido à sua natureza probabilística. Ao contrário da Regressão Linear, a Regressão Logística não tenta estimar o valor de y, mas sim a probabilidade do evento y = 1 ocorrer.

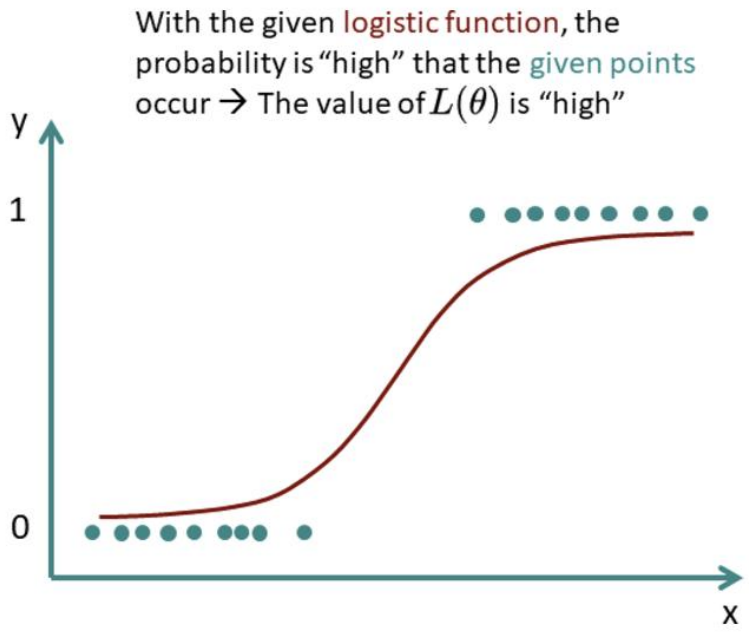
2.2 Algoritmos "Model-Based" – Regressão Logística

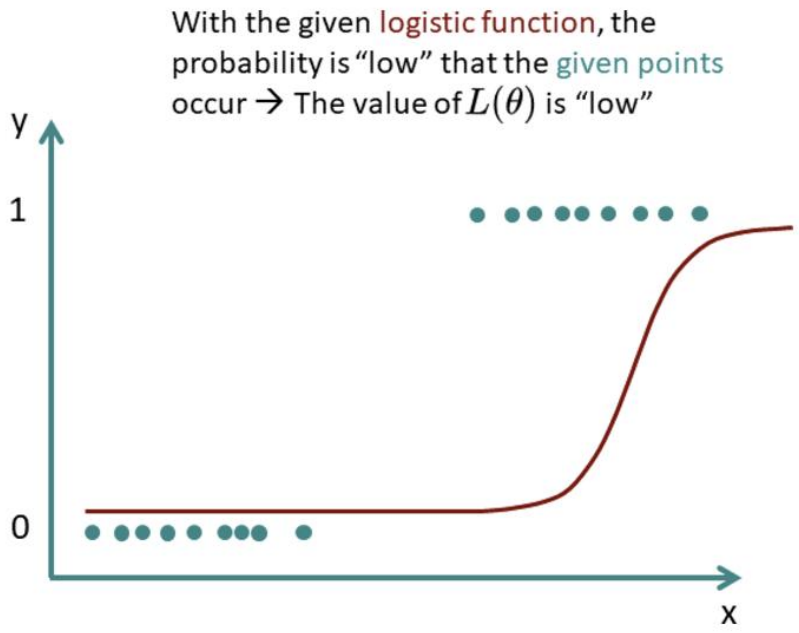
- Para estimar os parâmetros β_0 e β_1, não podemos recorrer à minimzação da soma dos desvios (Sum of Squared Residuals), visto não termos o termo do erro.
- Assim, recorremos ao método Maximum Likelihood, que procura responder à seguinte questão:

"Como podemos encontrar os melhores parâmetros do modelo para que as probabilidades previstas pelo modelo correspondam o mais possível aos resultados reais nos dados de treino?"

2.2 Algoritmos "Model-Based" – Regressão Logística

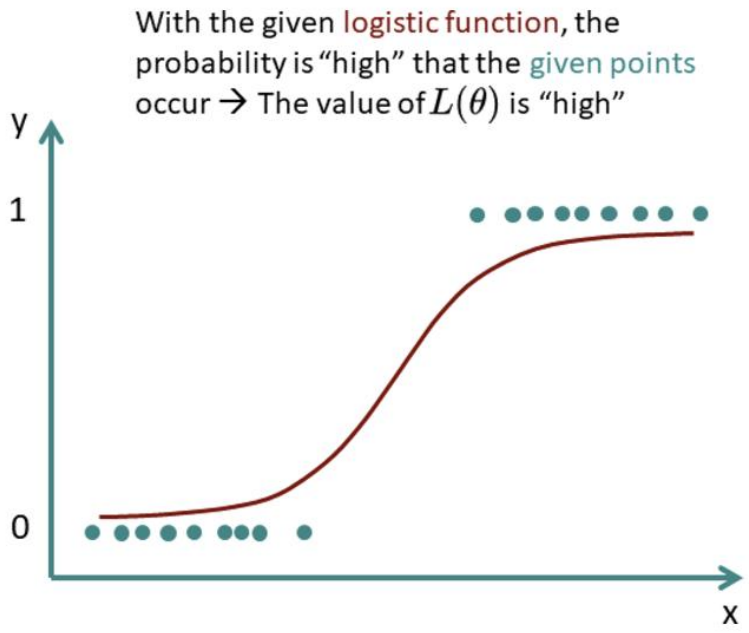
- Se definirmos $L(\theta) = L(\beta_0, \beta_1)$ como o grau de alinhamento entra as probabilidades previstas pelo modelo e os resultados reais nos dados de treino, podemos visualizar como muda a Regressão Logística para diferentes conjuntos de parâmetros θ:

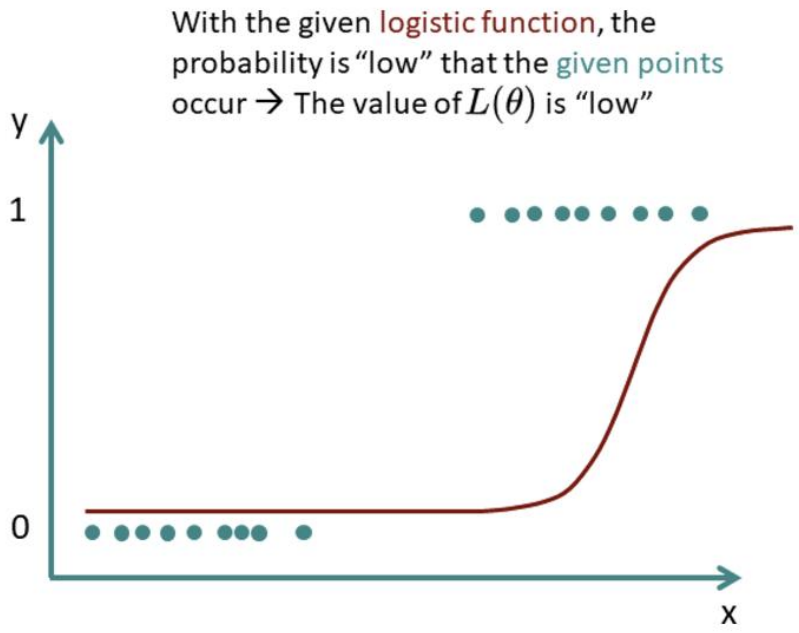




2.2 Algoritmos "Model-Based" – Regressão Logística

- O trabalho do Maxmium Likelihood é então calcular o melhor conjuntos de parâmetros θ, tal que as probabilidades previstas pelo modelo correspondam o mais possível aos resultados reais nos dados de treino (imagem da esquerda):





2.2 Algoritmos "Model-Based" – Regressão Logística

- No caso de existirem mais do que 1 variável independentes (cenário mais realístico), o processo é o mesmo, mas estimamos tantos parâmetros quanto o número de variáveis independentes (mais a ordenada na origem).

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 . x_1 + \beta_2 . x_2 + \cdots + \beta_n . x_n)}}$$

- Depois de estimar os parâmetros (com os dados de treino), apenas necessitamos de guardar esses parâmetros, e podemos estimar o valor da variável alvo para quaisquer valores das variáveis indepentendes, apenas multiplicando o parâmetro pelo valor da sua variável correspondente.

2.3 Algoritmos "Model-Based" – Regressão Lasso

- A regressão Lasso (ou "regularização l1") pertence a um conjunto de técnicas de regularização, aplicadas para reduzir o overfitting.
- Outra técnica de regularização bastante conhecida é a regressão Ridge (ou "regularização l2".

2.3 Algoritmos "Model-Based" – Regressão Lasso

- A regressão Lasso (ou "regularização l1") pertence a um conjunto de técnicas de regularização, aplicadas para reduzir o overfitting.
- Outra técnica de regularização bastante conhecida é a regressão Ridge (ou "regularização l2".
- A regressão Lasso é fundamentalmente uma extensão da regressão linear (podendo ser aplicada também à regressão logística):
  - Na regressão linear pretendemos minimizar $\sum \epsilon^2 = \sum (y - \hat{y})^2$;
  - Na regressão Lasso pretendemos minimizar $\sum \epsilon^2 + \lambda \times \sum |\beta_i|$, onde λ controla a intensidade da penalização.

2.3 Algoritmos “Model-Based” – Regressão Lasso

- A intuição é a seguinte: o termo de penalização “l1” na regressão lasso reduz a zero os coeficientes das variáveis menos significativas.
- Assim, as variáveis com coeficientes nulos podem ser eliminadas do modelo. A regressão Lasso torna o modelo mais simples e menos propenso a overfitting.
- Como é possível entender, esta técnica será bastanté útil no processo de “feature selection”.

2.4 Algoritmos "Model-Based" – Regressão Ridge

- A regressão Ridge (ou "regularização l2") é semelhante à regressão Lasso, mas o termo de penalização em vez de $\lambda \times \sum|\beta_i|$ é $\lambda \times \sum \beta_i^2$.
- No entanto, a regressão Ridge é menos agressiva, pelo que, por construção não reduz para 0 os coeficientes das variáveis menos significativas (ver https://www.youtube.com/watch?v=Xm2C_gTAl8c para melhor intuição) .
- Assim, é mais adequado para situações em que todas as variáveis independentes são potencialmente relevantes e o objetivo é apenas reduzir o overfitting em vez de selecioanr variávies.

Obrigado!