

CURSO BÁSICO DE  
INTELIGÊNCIA  
ARTIFICIAL E  
BATE-PAPO COM  
CONVIDADOS  
ESPECIAIS



# INTELIGÊNCIA ARTIFICIAL PARA TODOS

DE 08/06 A 12/08



COM OS PROFESSORES DO  
LABORATÓRIO ARIA/UFPB:  
TELMO FILHO, THAÍS  
GAUDENCIO E YURI  
MALHEIROS

 Centro de Informática  
UFPB

 Departamento de  
ESTATÍSTICA

 artificial  
intelligence  
applications

- CURSO SEM PRÉ-REQUISITOS
- [HTTP://ARIA.CI.UFPB.BR/IAPARATODOS](http://aria.ci.ufpb.br/iaparatos)
- INSCRIÇÃO PARA CERTIFICADO - DE 01/06  
A 07/06: [HTTP://BIT.LY/SIGEVENTOS](http://bit.ly/sigeventos)
- ENCONTROS: SEGUNDAS E QUARTAS
- HORÁRIO: 19:00 ÀS 20:00



[Início](#) [Sobre](#) [Projetos](#) [Membros](#) [Parceiros](#) [Publicações](#) [Contato](#)

## LABORATÓRIO DE APLICAÇÕES EM INTELIGÊNCIA ARTIFICIAL

As experiências definem a aprendizagem. Assim, o ARIA constrói experiências para máquinas e para pessoas, formando especialistas na área de inteligência artificial e ciência de dados, desenvolvendo aplicações e pesquisando seus métodos.

[SAIBA MAIS](#)

**SE INSCREVE E JÁ APERTA NO SININHO, QUE VOCÊS PASSAM A RECEBER AS NOTIFICAÇÕES.**

**NOSSOS ENCONTROS DURARÃO 1 HORA E, ASSIM QUE POSSÍVEL, DEIXAREMOS OS VÍDEOS GRAVADOS NO CANAL.**

**NÃO PRECISA SE PREOCUPAR EM ESTAR LIGADO ÀS 19:00, MAS ESTANDO, ROLA TIRAR DÚVIDA E PARTICIPAR, O QUE JÁ DEIXA A AULA MAIS ANIMADA.**

**SOBRE O MATERIAL DE ACOMPANHAMENTO: O ALUNO PRECISA SE LOGAR EM:  
CLASSROOM.GOOGLE.COM**

**DEPOIS, CLICAR EM PARTICIPAR DA TURMA (ÍCONE COM UM MAIS +)**

**POR FIM, ENTRAR COM O CÓDIGO DA TURMA: PXV3ANW**

**TAMBÉM EM: [HTTPS://ARIA.CI.UFPB.BR/IA-PARA-TODOS-MATERIAL/](https://aria.ci.ufpb.br/ia-para-todos-material/)**

**ESPERAMOS QUE VOCÊS REALMENTE CURTAM O CURSO E APROVEITEM-NO AO MÁXIMO. NÃO DEIXEM DE INTERAGIR CONOSCO, TAMBÉM, POR E-MAIL OU MENSAGEM NO NOSSO INSTAGRAM (@APRENDIZAGEMDEMAQUINA)**



# Aprendizagem Supervisionada

O algoritmo de aprendizagem recebe um conjunto com exemplos para os quais os rótulos são conhecidos. Esses rótulos podem ser associados a diferentes tarefas:

Classificação = Categorização em uma ou mais classes

Regressão = Estimativa de valores para uma ou mais variáveis de interesse



Gato

Cachorro

Gato



?



# Classificação

Como visto na aula anterior, os exemplos de um conjunto de treinamento são representados por vetores com um certo número de variáveis explicativas

Na tarefa de classificação, espera-se que exemplos de classes diferentes tenham diferentes padrões de valores dessas variáveis

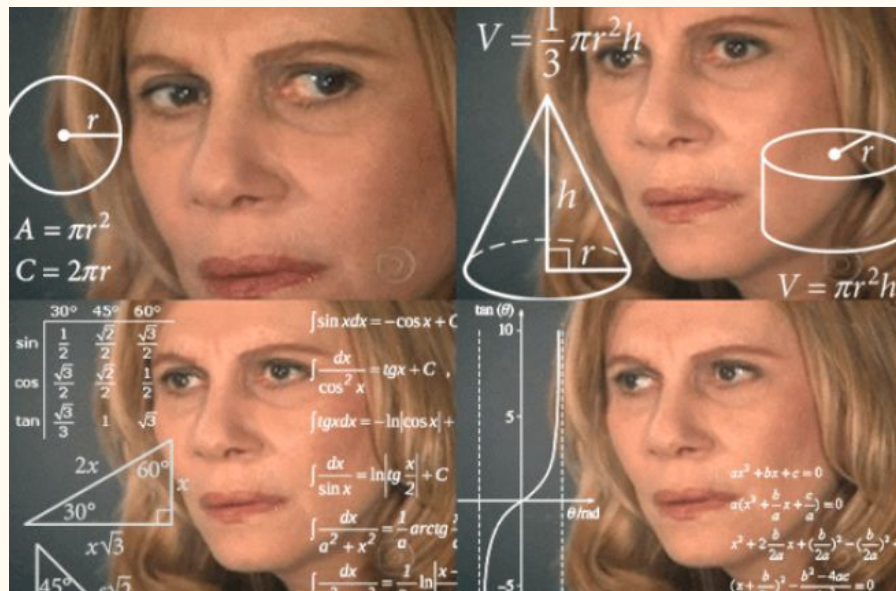
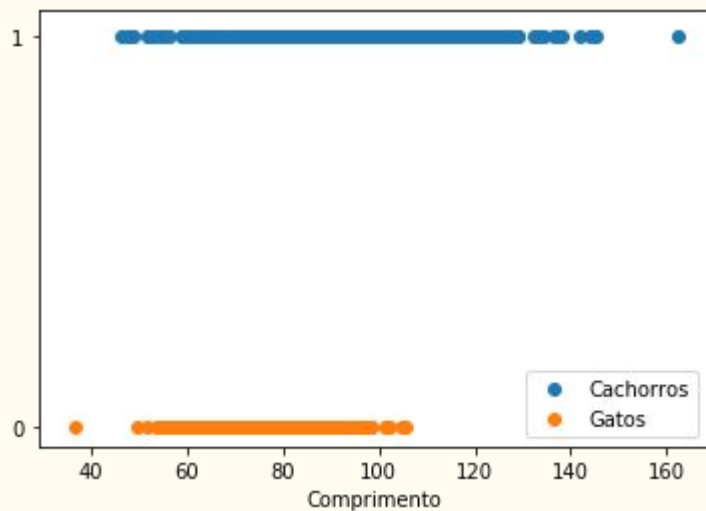
# Exemplo de conjunto de dados

O conjunto abaixo contém dados fictícios de cães e gatos:

	<b>Comprimento (cm)</b>	<b>Altura (cm)</b>	<b>Peso (kg)</b>	<b>Classe</b>
1	89,64	44,23	20,39	Cachorro
2	69,27	43,06	15,03	Cachorro
3	70,52	28,30	12,32	Cachorro
4	88,28	49,45	15,81	Cachorro
...	...	...	...	...
996	87,46	18,67	4,21	Gato
997	69,98	23,90	5,57	Gato
998	85,52	23,87	6,33	Gato
999	88,44	23,76	5,63	Gato

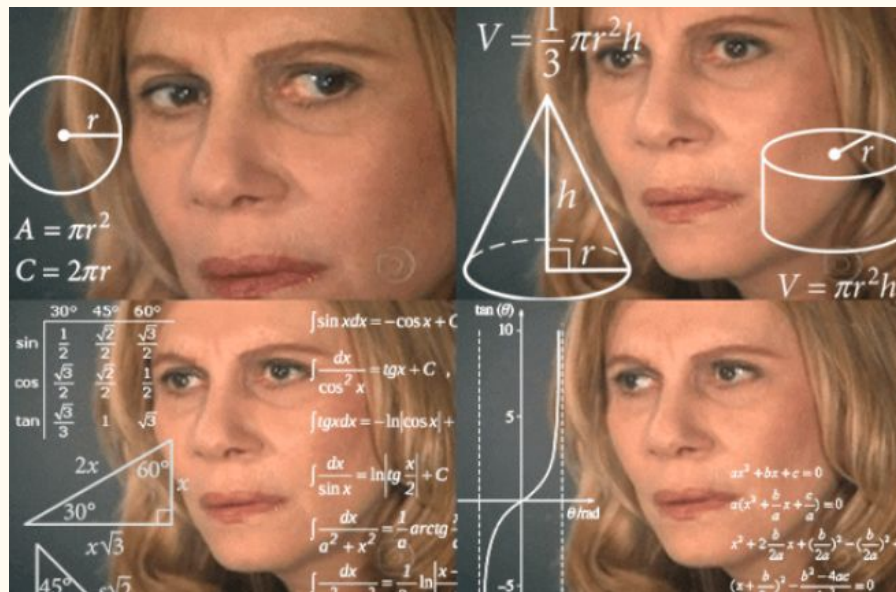
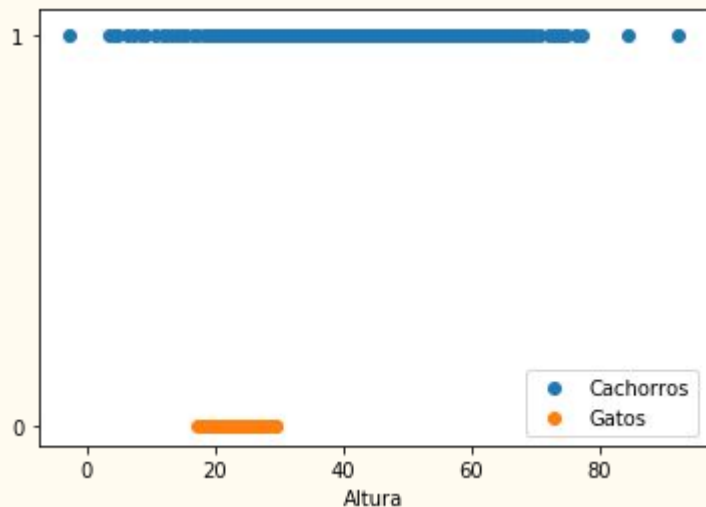
# Como nós humanos faríamos essa tarefa?

Uma forma intuitiva de categorizar novos exemplos como cães e gatos é observar as variáveis que os descrevem



# Como nós humanos faríamos essa tarefa?

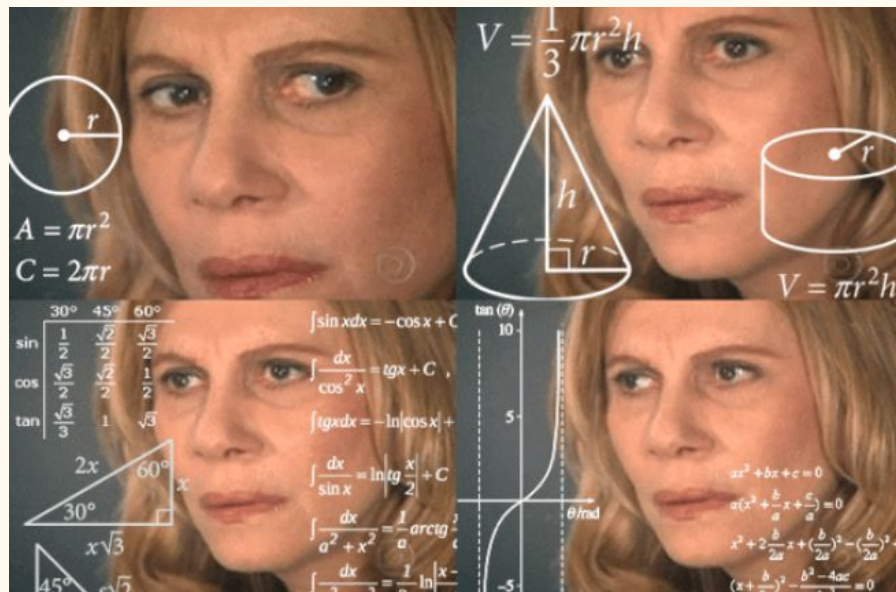
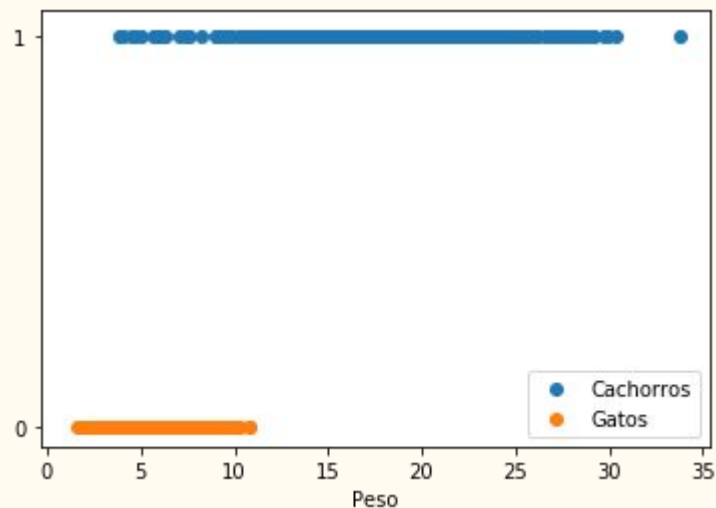
Uma forma intuitiva de categorizar novos exemplos como cães e gatos é observar as variáveis que os descrevem





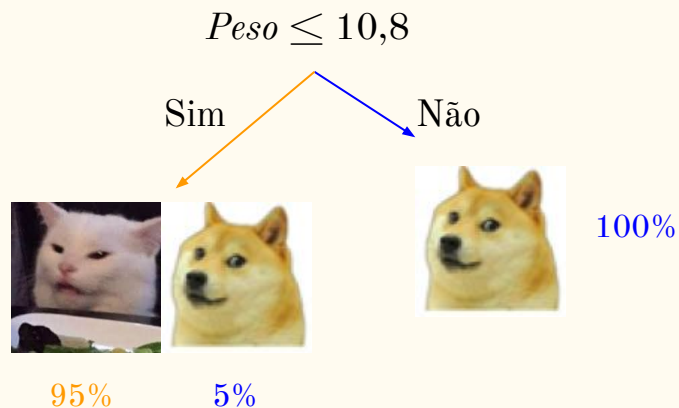
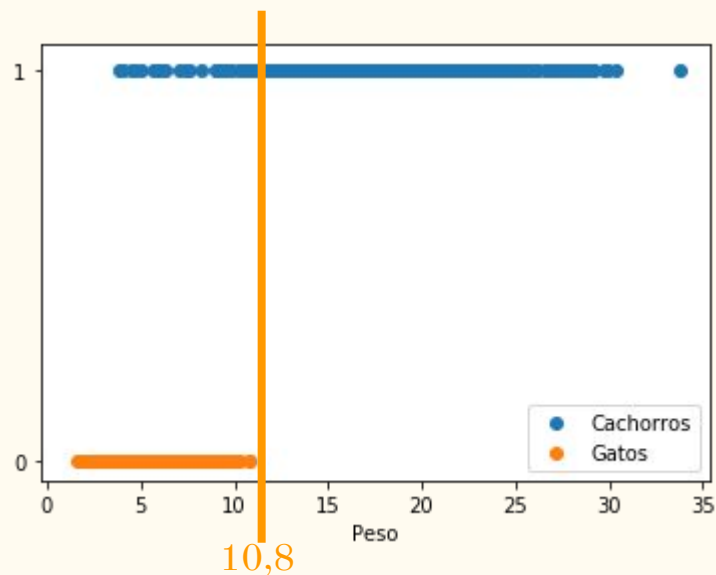
# Como nós humanos faríamos essa tarefa?

Uma forma intuitiva de categorizar novos exemplos como cães e gatos é observar as variáveis que os descrevem



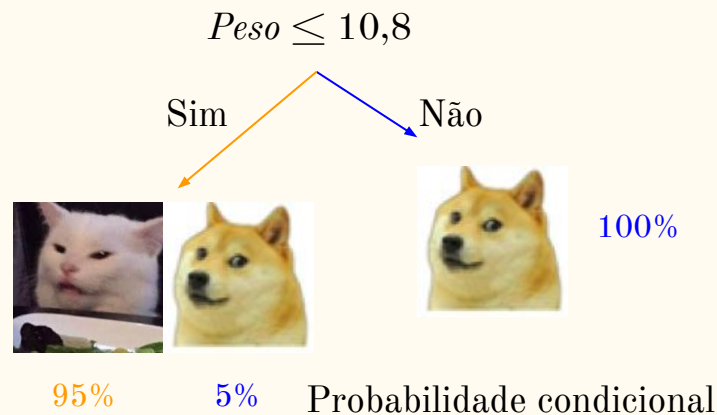
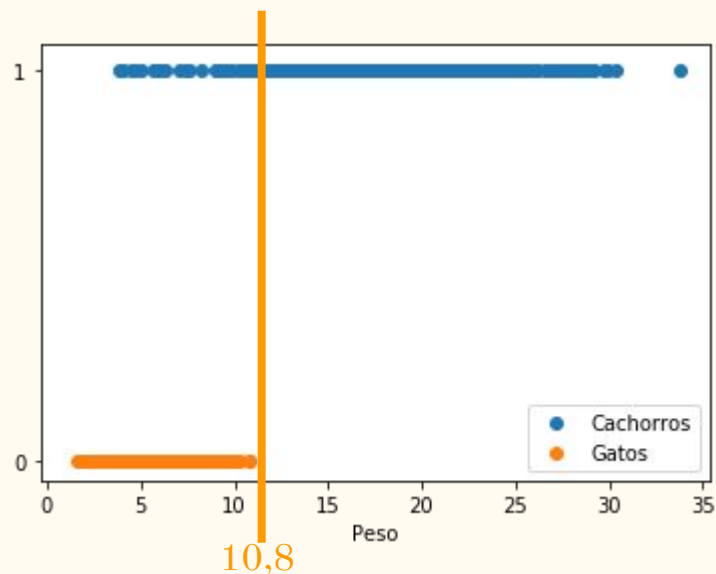
# Como nós humanos faríamos essa tarefa?

Podemos definir um ponto de corte em cima da variável *Peso*



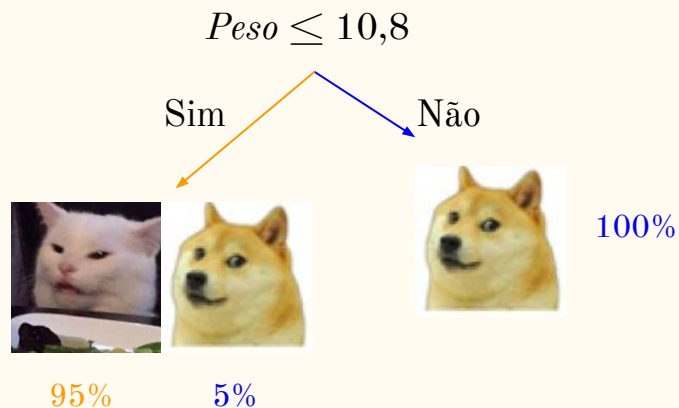
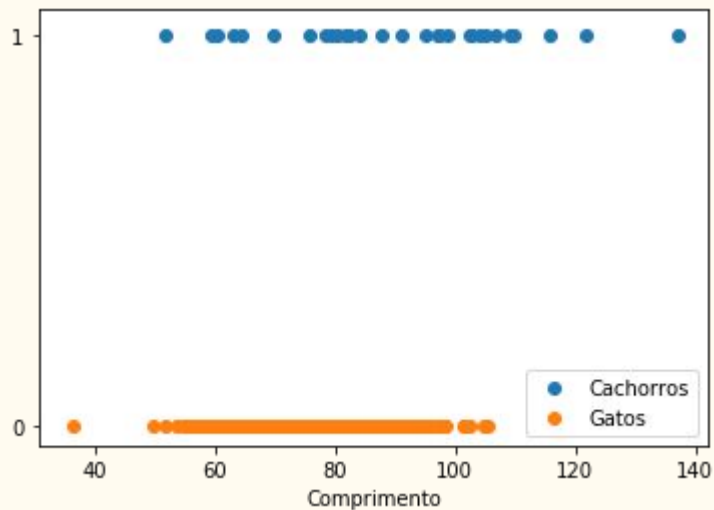
# Como nós humanos faríamos essa tarefa?

Podemos definir um ponto de corte em cima da variável *Peso*



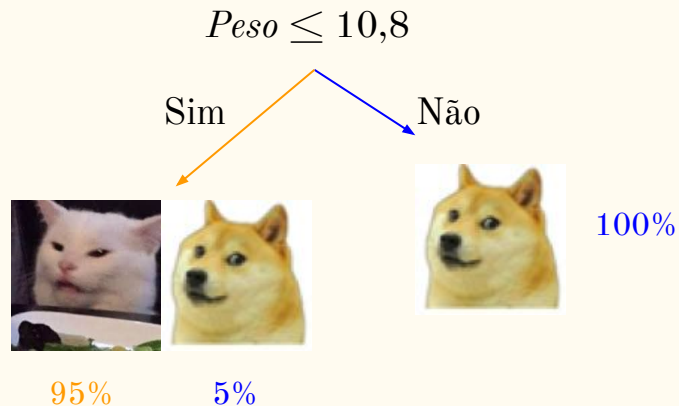
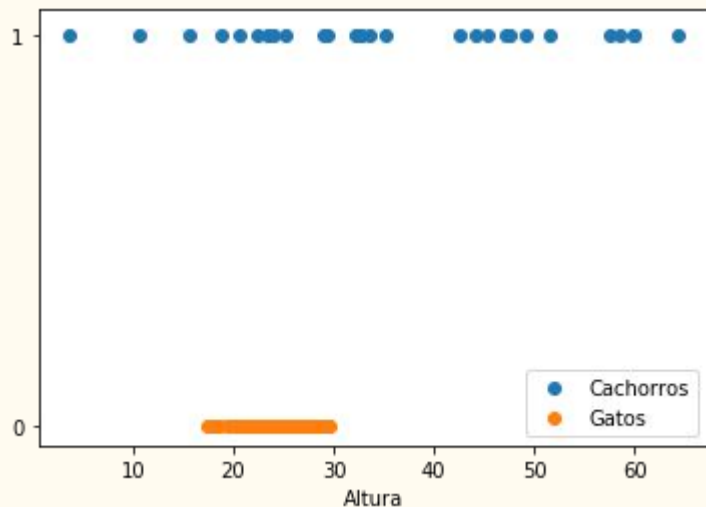
# Como nós humanos faríamos essa tarefa?

Podemos agora tentar diferenciar os animais que têm menos de 10,8kg usando as outras duas variáveis



# Como nós humanos faríamos essa tarefa?

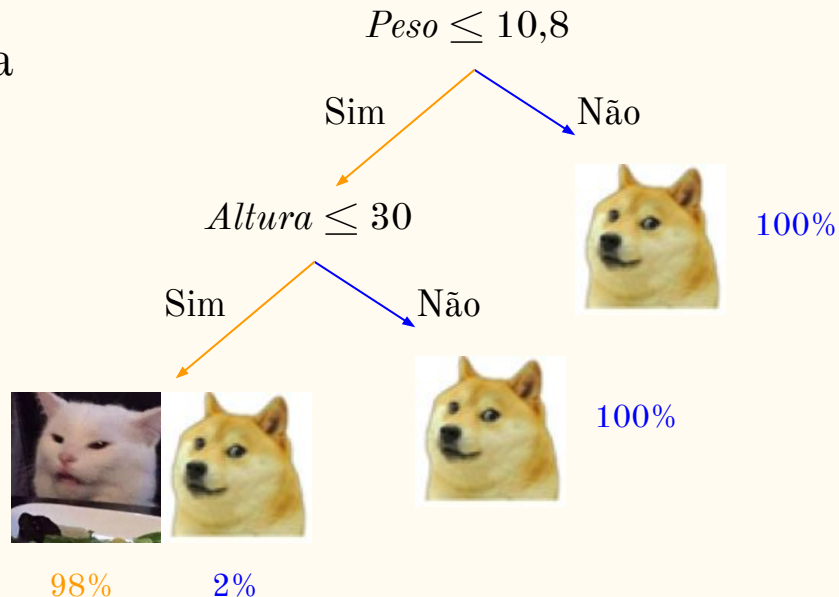
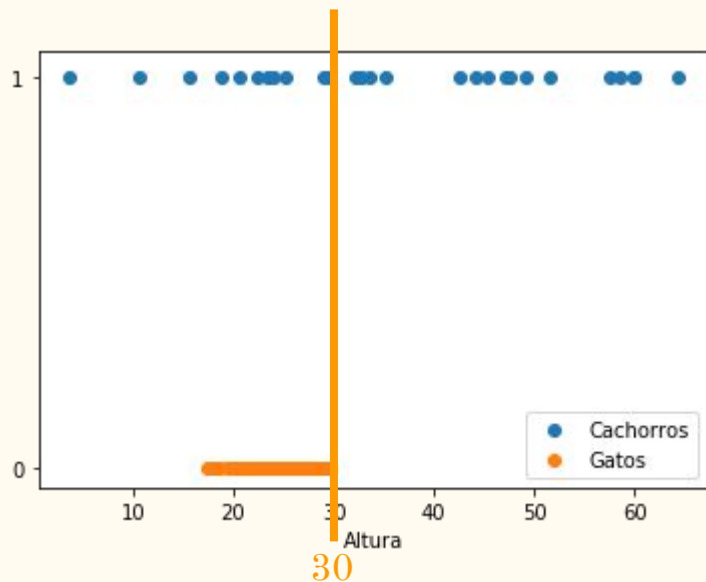
Podemos agora tentar diferenciar os animais que têm menos de 10,8kg usando as outras duas variáveis





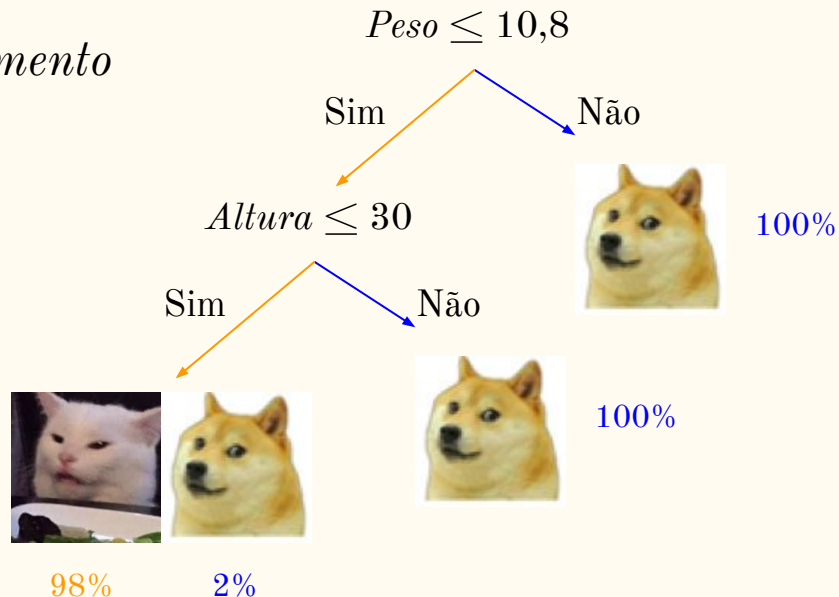
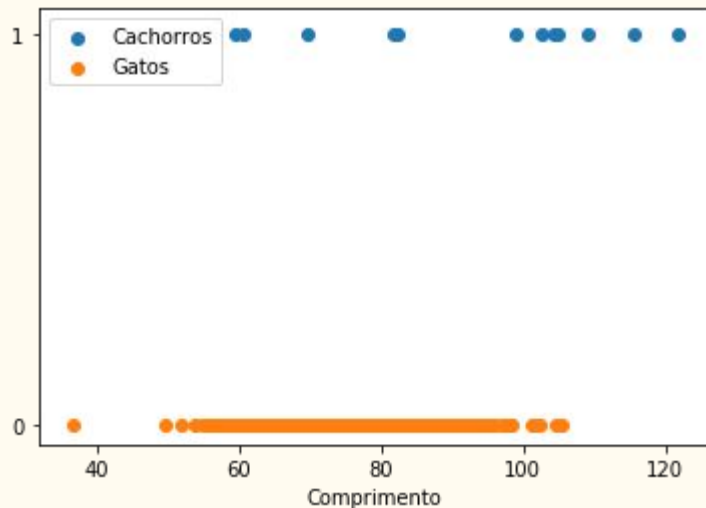
# Como nós humanos faríamos essa tarefa?

Fazemos o próximo ponto de corte usando a variável *Altura*



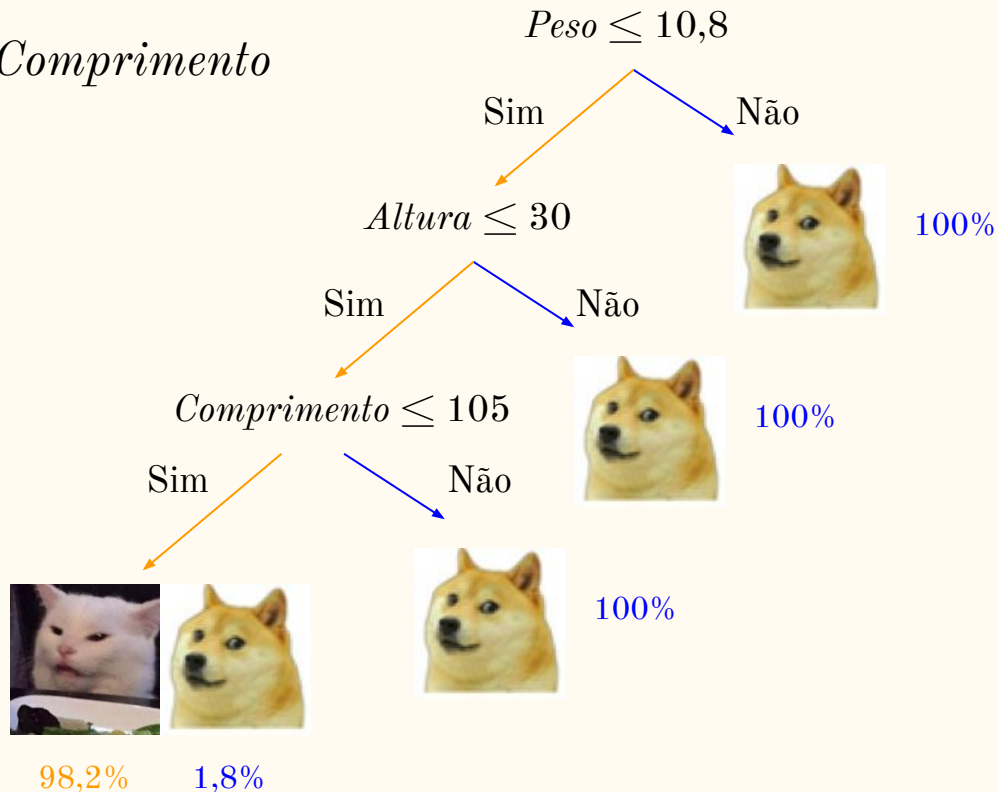
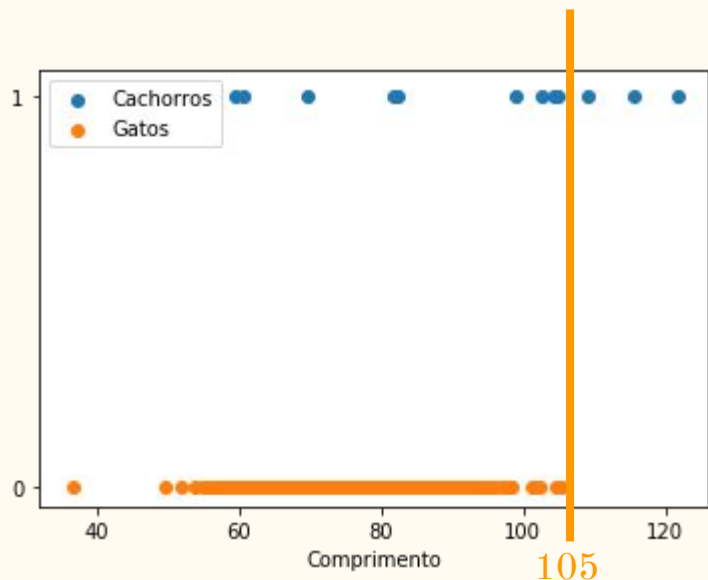
# Como nós humanos faríamos essa tarefa?

Por último, ficamos com a variável *Comprimento*



# Como nós humanos faríamos essa tarefa?

Por último, ficamos com a variável *Comprimento*



# Seguimos um algoritmo de forma natural

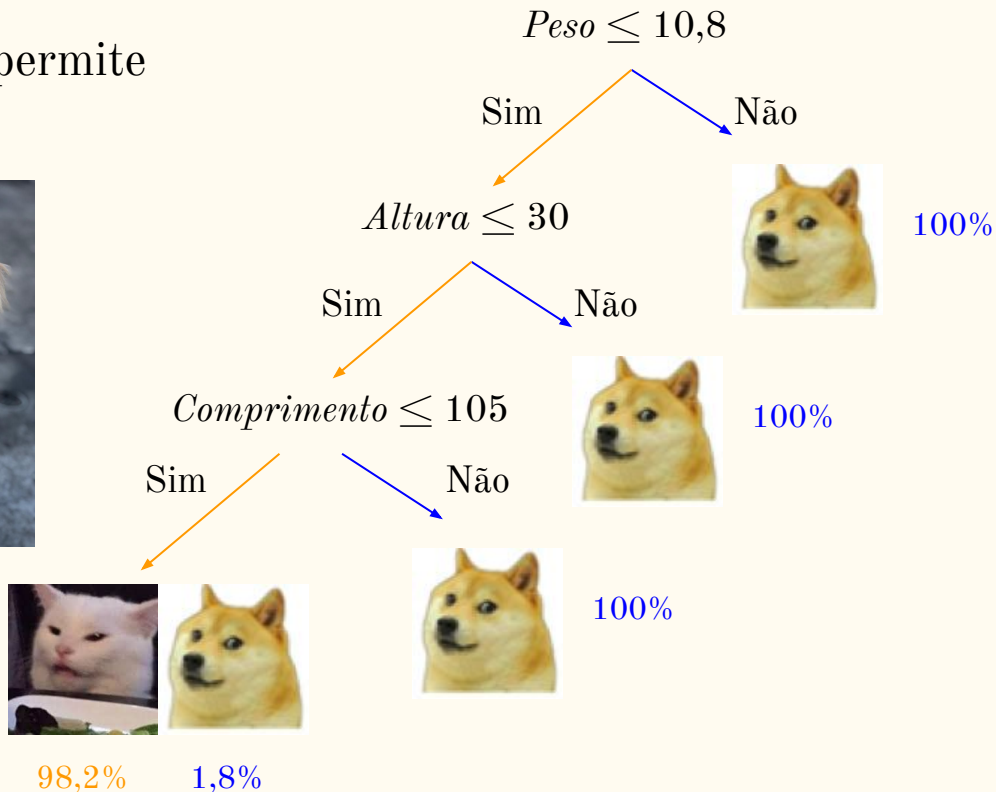
E agora temos um modelo que nos permite categorizar novos exemplos

Exemplo:

Peso: 10,7kg

Altura: 46cm

Comprimento: 110cm



# Seguimos um algoritmo de forma natural

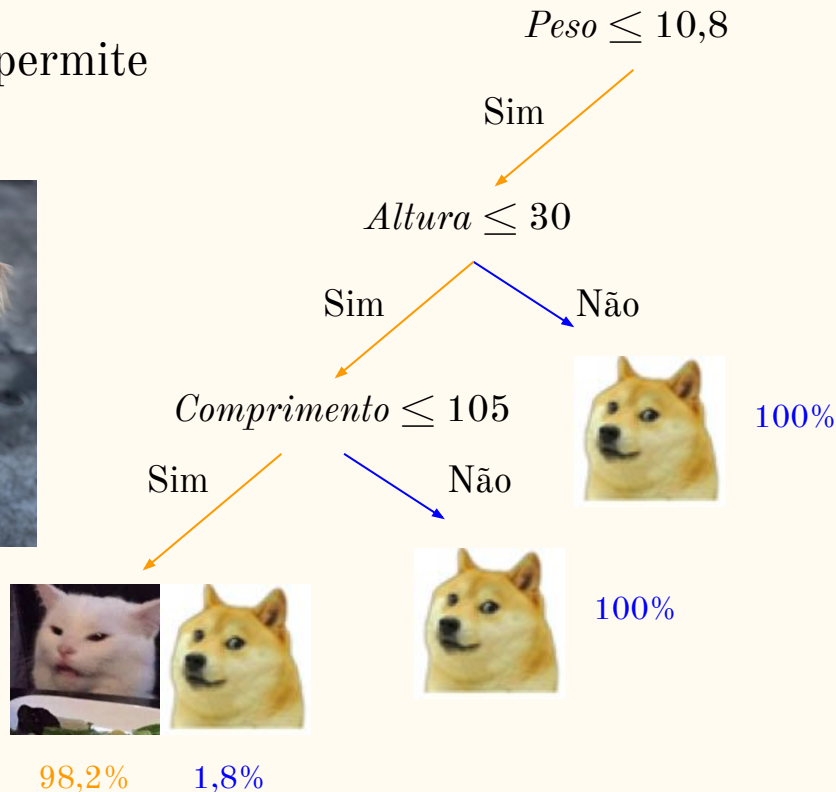
E agora temos um modelo que nos permite categorizar novos exemplos

Exemplo:

Peso: 10,7kg

Altura: 46cm

Comprimento: 110cm





# Seguimos um algoritmo de forma natural

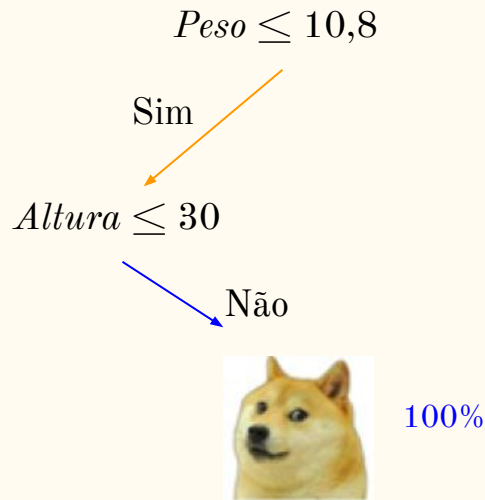
E agora temos um modelo que nos permite categorizar novos exemplos

Exemplo:

Peso: 10,7kg

Altura: 46cm

Comprimento: 110cm



# Árvore de Decisão

# O que é árvore de decisão?

Um modelo baseado em fluxogramas

Muito usado devido a sua simplicidade e interpretabilidade



# O que é árvore de decisão?

Sua construção segue um algoritmo igual ao que acabamos de desenvolver

1. Escolha variável que resulta em maior “pureza”
  - a. Escolha um ponto de corte
2. Repita o passo 1 até que todas as “folhas” sejam puras



```
from sklearn.tree import DecisionTreeClassifier
```

```
X = data[['Comprimento', 'Altura', 'Peso']]
```

```
y = data['Classe']
```

```
dt = DecisionTreeClassifier(max_depth=3)
```

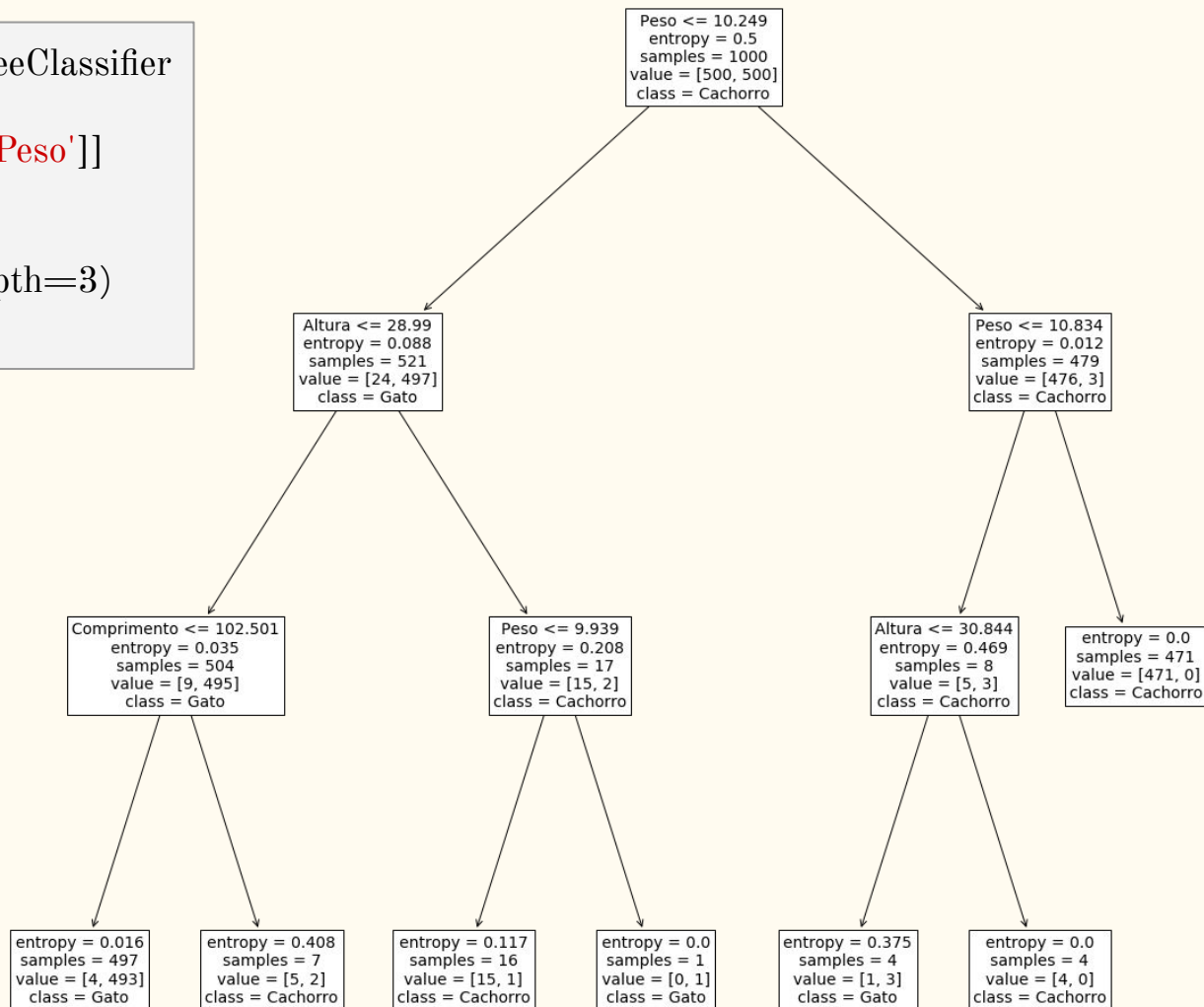
```
dt.fit(X, y)
```

Um detalhe legal das árvores de decisão é que elas podem indicar a **importância** das variáveis:

Comprimento: 0,01

Altura: 0,06

Peso: 0,93



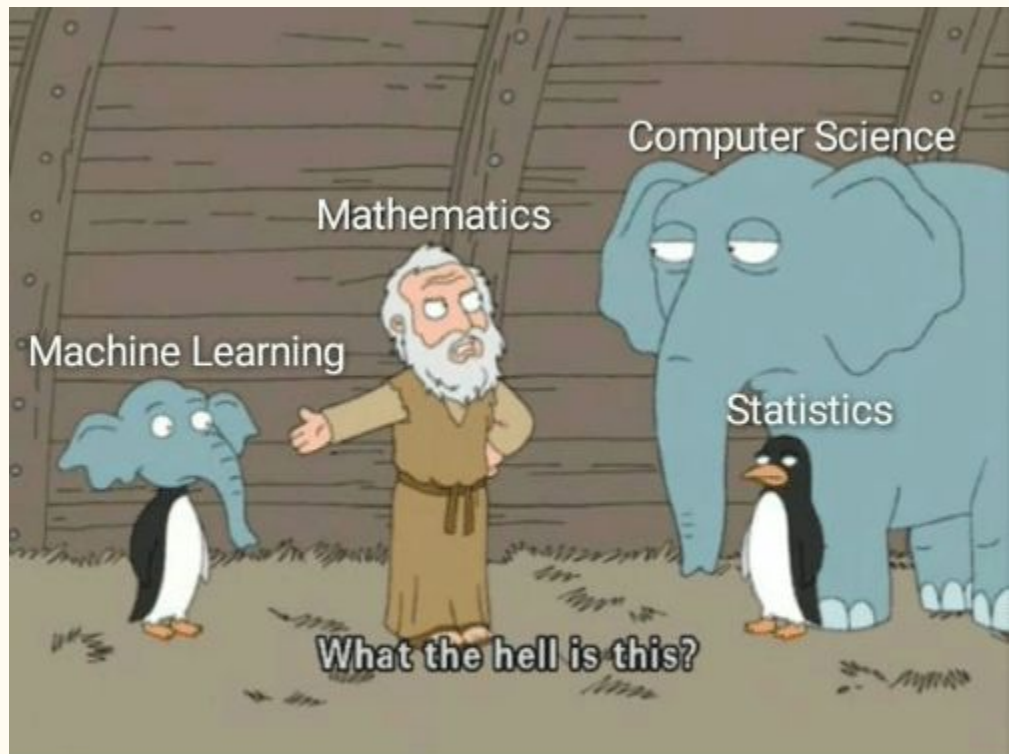


Árvore de decisão é apenas um dos vários algoritmos de classificação disponíveis

Obs: também existem árvores de decisão para regressão

Cada algoritmo busca aprender de formas diferentes

No final das contas, isso equivale a misturar ciência da computação, estatística e matemática de maneiras diferentes



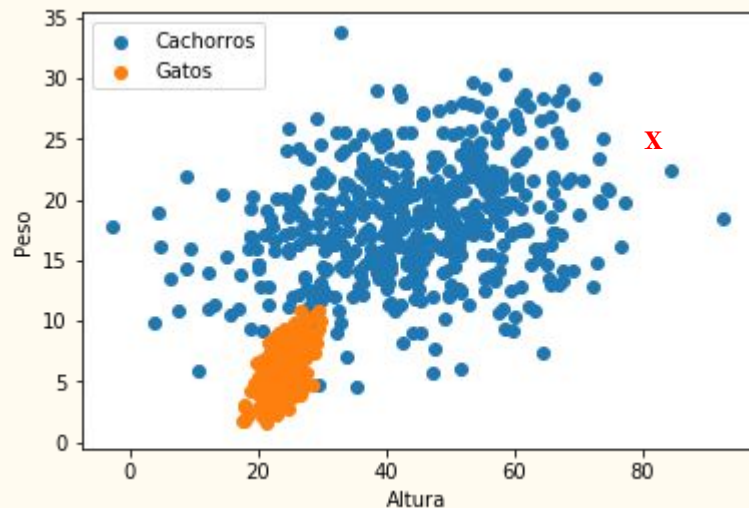
Machine learning

k-vizinhos mais  
próximos

# Diga-me com quem andas...

Observe a posição do novo elemento marcado por um **x**

Qual deve ser a classe atribuída a ele?

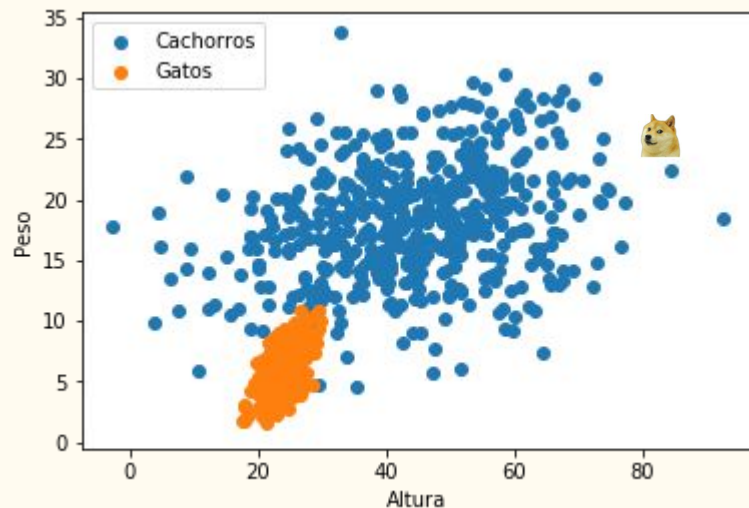


# Que te direi quem és

Observe a posição do novo elemento marcado por um **x**

Qual deve ser a classe atribuída a ele?

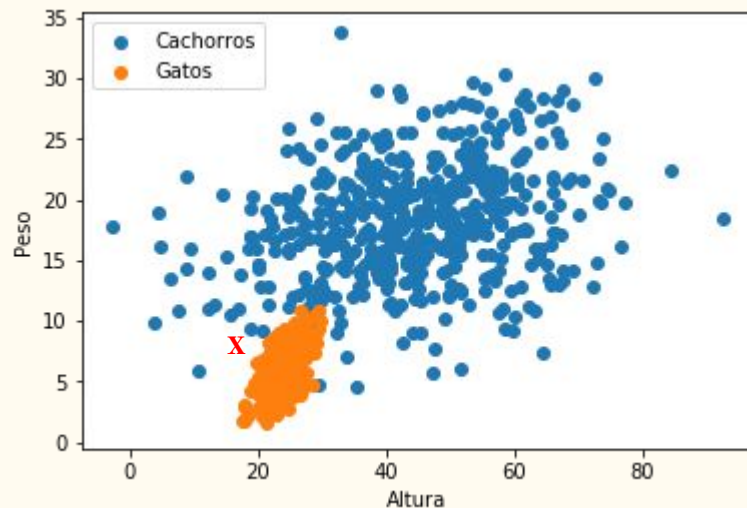
Os exemplos mais próximos a ele são todos cachorros, então parece seguro que ele também será



# Diga-me com quem anda...

Observe a posição do novo elemento marcado por um **x**

Qual deve ser a classe atribuída a ele?



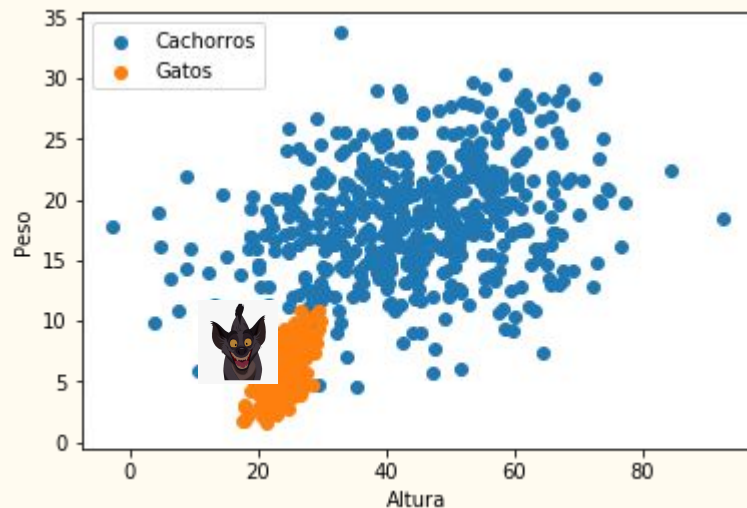


# Diga-me com quem andas...

Observe a posição do novo elemento marcado por um **x**

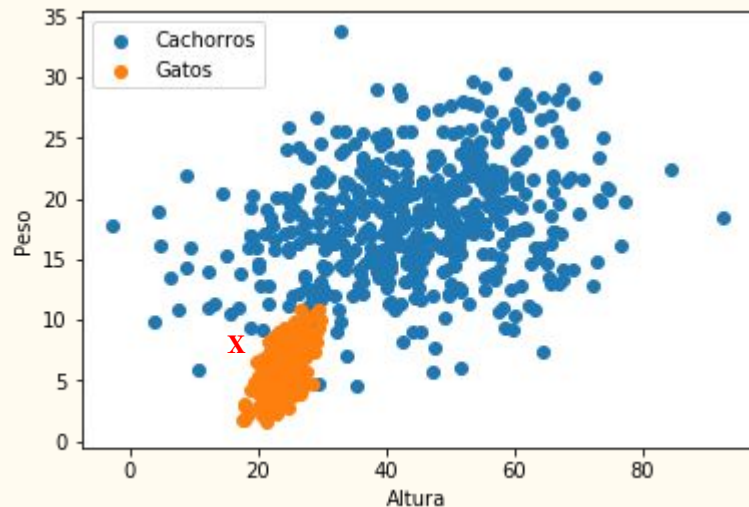
Qual deve ser a classe atribuída a ele?

Depende...



# k-vizinhos mais próximos

Algoritmo que surgiu da nossa tendência de categorizar coisas de acordo com sua semelhança para o que conhecemos



# k-vizinhos mais próximos

Um “modelo” k-NN simplesmente memoriza os dados de treinamento

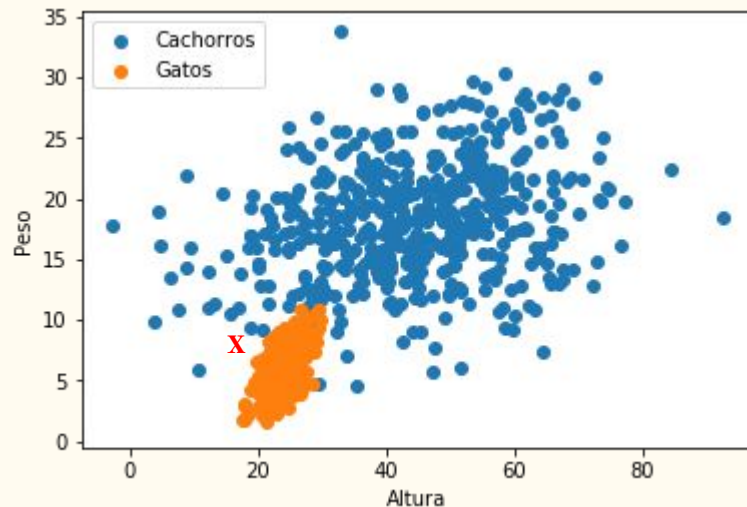
```
from sklearn.neighbors import KNeighborsClassifier
```

```
X = data[['Altura', 'Peso']]
```

```
y = data['Classe']
```

```
knn = KNeighborsClassifier()
```

```
knn.fit(X, y)
```

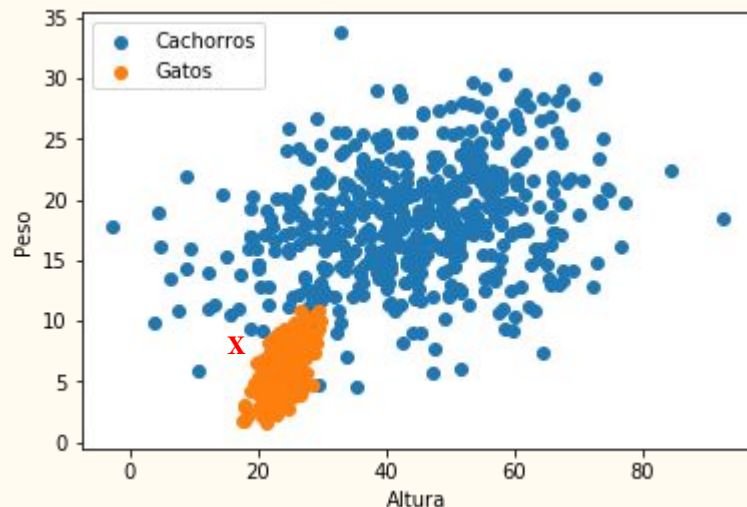


# k-vizinhos mais próximos

Para classificar um novo exemplo, o k-NN calcula sua distância para os dados conhecidos e toma a decisão baseado nas classes dos “ $k$ ” mais próximos

Comumente usa-se a distância Euclidiana:

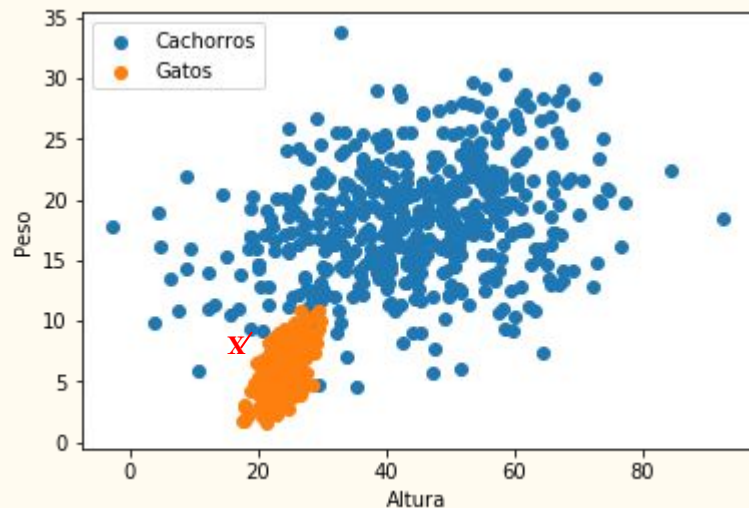
$$d(\vec{x}, \vec{z}) = \sqrt{(x_{Altura} - z_{Altura})^2 + (x_{Peso} - z_{Peso})^2}$$



# k-vizinhos mais próximos

A decisão tomada pelo k-NN depende do valor escolhido de  $k$

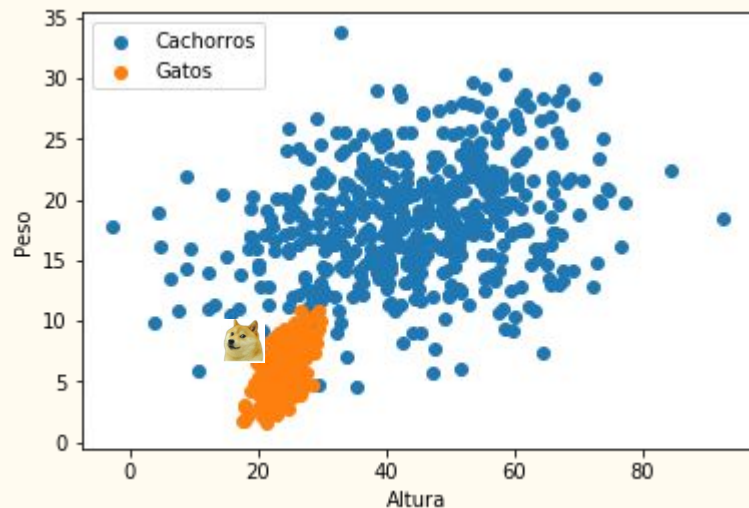
Para  $k = 1$ :



# k-vizinhos mais próximos

A decisão tomada pelo k-NN depende do valor escolhido de  $k$

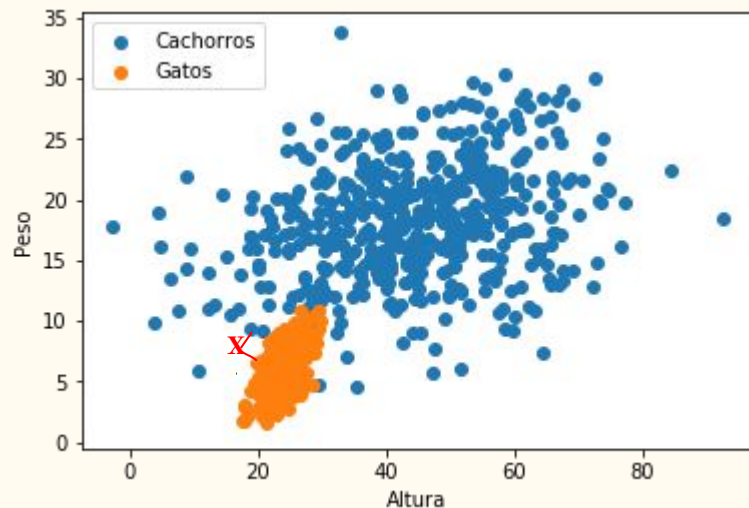
Para  $k = 1$ :



# k-vizinhos mais próximos

A decisão tomada pelo k-NN depende do valor escolhido de  $k$

Para  $k = 2$ :



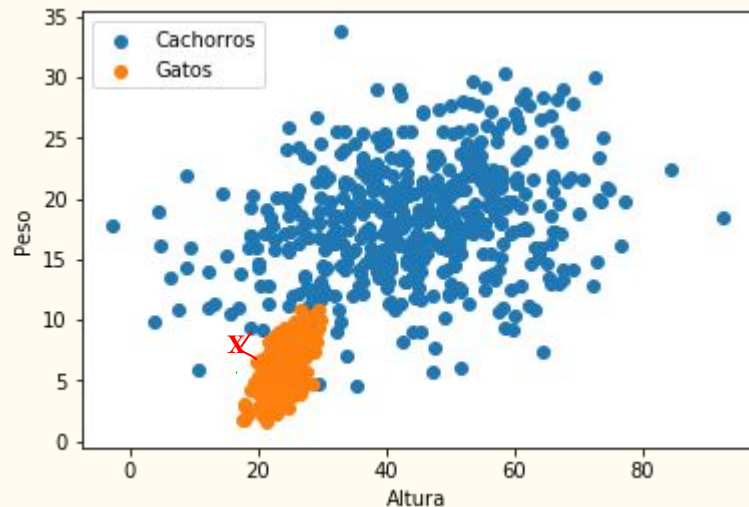
# k-vizinhos mais próximos

A decisão tomada pelo k-NN depende do valor escolhido de  $k$

Para  $k = 2$ :

Temos um empate entre Gato e Cachorro

Podemos escolher aleatoriamente ou ponderar pela distância





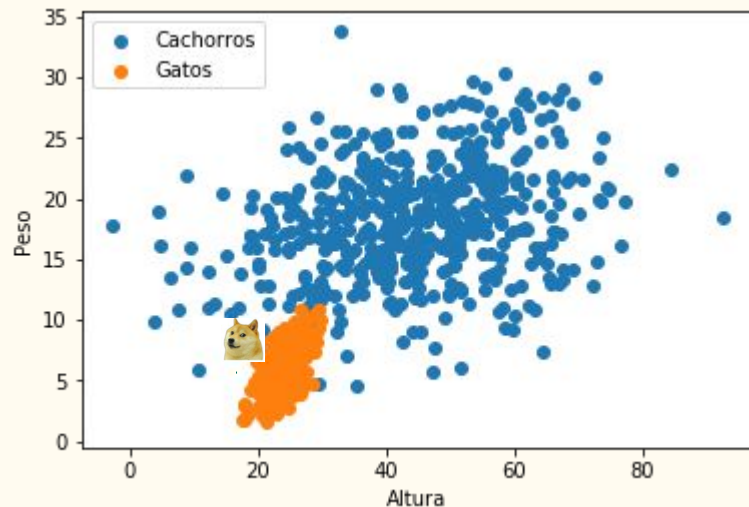
# k-vizinhos mais próximos

A decisão tomada pelo k-NN depende do valor escolhido de  $k$

Para  $k = 2$ :

Temos um empate entre Gato e Cachorro

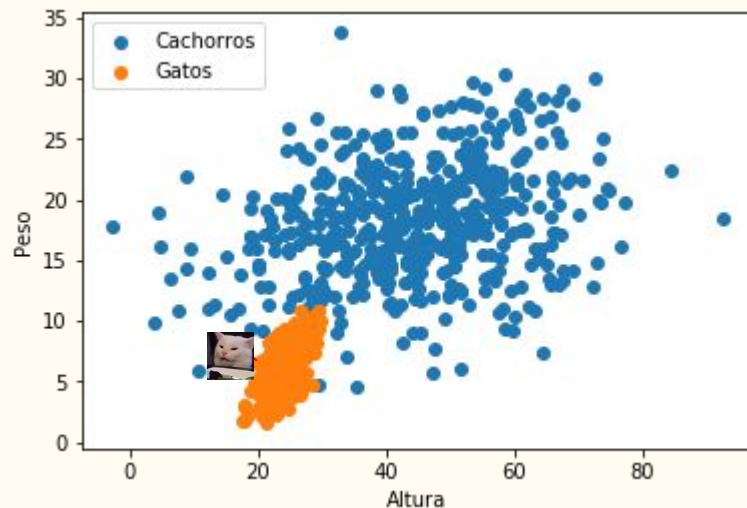
Podemos escolher aleatoriamente ou ponderar pela distância



# k-vizinhos mais próximos

A decisão tomada pelo k-NN depende do valor escolhido de  $k$

Para  $k > 10$ :



# Regressão

**BRACE YOURSELF**

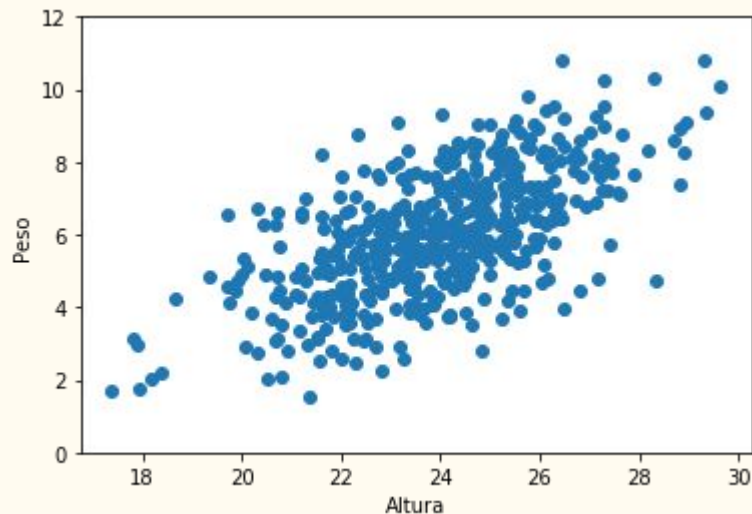


**REGRESSION IS COMING**

# Quando usamos regressão?

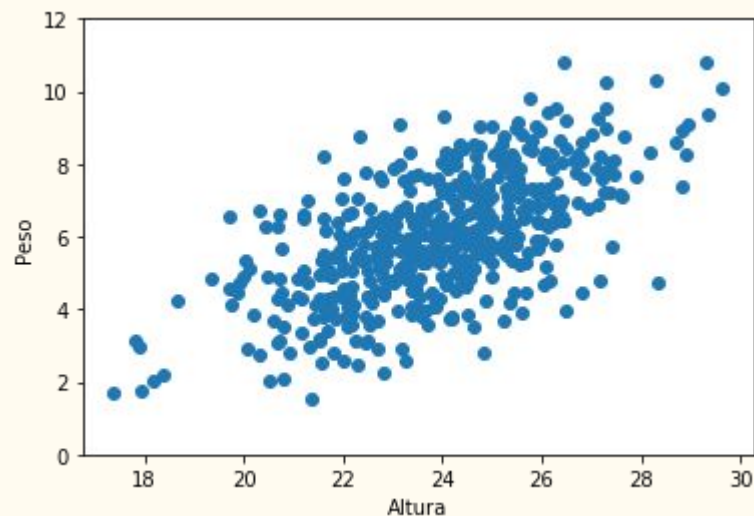
Regressão é usada quando queremos estimar um valor numérico

A ideia é encontrar uma **função** que receba valores de uma ou mais variáveis e dê como resultado valores de uma ou mais variáveis desejadas



# Quando usamos regressão?

Regressão também pode ser usada para determinar se as variáveis explicativas realmente têm alguma relação (linear) com a variável alvo



# Regressão linear

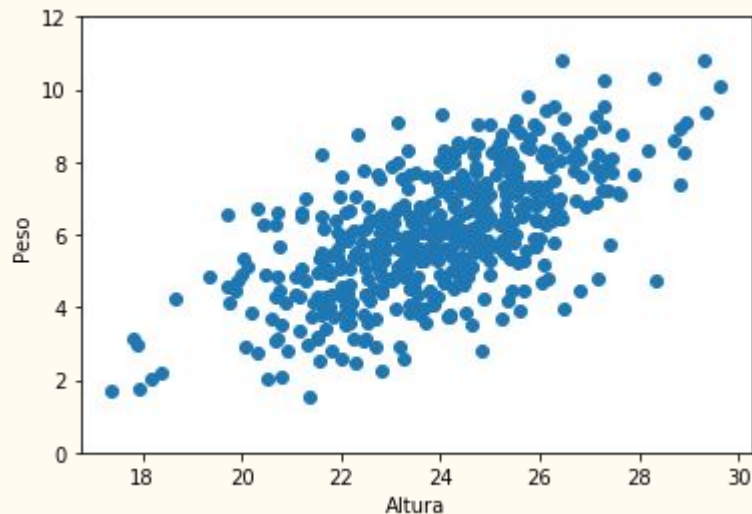
Modelo que assume que existe uma relação linear entre a variável alvo ( $Y=Peso$ ) e a variável explicativa ( $X=Altura$ )

$$Peso = \beta_0 + \beta_1 * Altura + \epsilon$$

$\beta_1$  Representa o quanto o valor de *Peso* aumenta a cada incremento de 1 unidade na *Altura*

$\beta_0$  Representa o quanto o valor de *Peso* quando *Altura* = 0

$\epsilon$  Chamado de erro ou ruído, representa todos os outros fatores que podem influenciar o valor de *Peso*

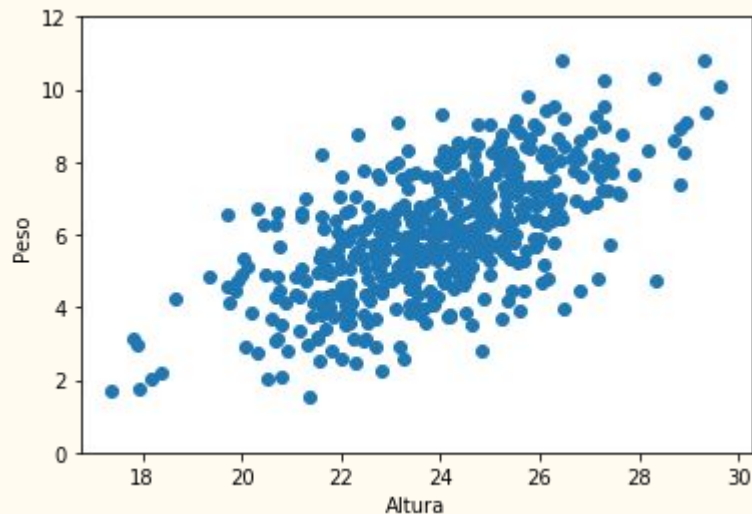


# Regressão linear

Modelo que assume que existe uma relação linear entre a variável alvo ( $Y=Peso$ ) e a variável explicativa ( $X=Altura$ )

$$Peso = \beta_0 + \beta_1 * Altura + \epsilon$$

O modelo de regressão linear é induzido usando o método dos **mínimos quadrados**, cujo objetivo é encontrar os coeficientes  $\beta_0$  e  $\beta_1$  que minimizam o erro  $\epsilon$  simultaneamente para todos os exemplos

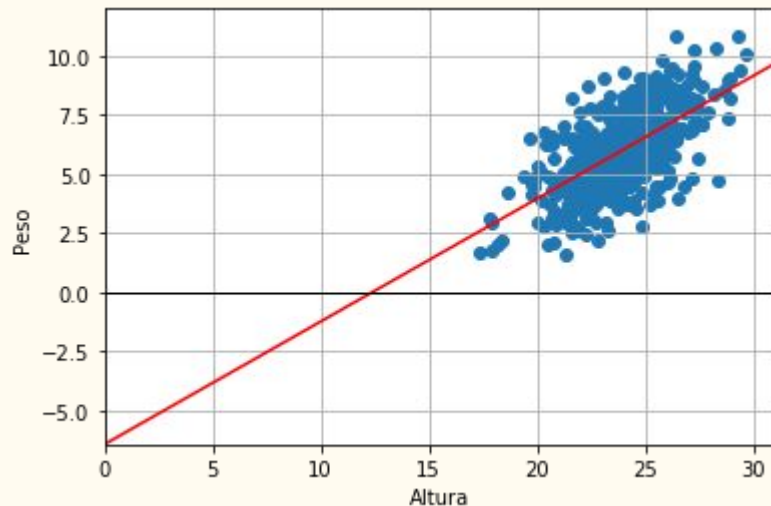




# Regressão linear

Modelo que assume que existe uma relação linear entre a variável alvo ( $Y=Peso$ ) e a variável explicativa ( $X=Altura$ )

$$Peso = -6,44 + 0,52 * Altura$$



# Regressão linear múltipla

Modelo que assume que existe uma relação linear entre a variável alvo ( $Y=Peso$ ) e duas ou mais variáveis explicativas, como  $X1=Altura$  e  $X2=Comprimento$

$$Peso = \beta_0 + \beta_1 * Altura + \beta_2 * Comprimento + \epsilon$$

Os coeficientes da equação de regressão são obtidos pelo método dos mínimos quadrados usando a seguinte equação:

$$\vec{\hat{\beta}} = (X^T X)^{-1} X^T Y \quad \text{Participação especial: Álgebra linear}$$

$$Peso = -7,46 + 0,52 * Altura + 0,01 * Comprimento$$

```
from sklearn.linear_model import LinearRegression
```

```
lr = LinearRegression()  
lr.fit(X, Y)
```

# Predição de um novo valor

*Altura* = 28 cm

*Comprimento* = 87 cm

$$Peso = -7,46 + 0,52 * 28 + 0,01 * 87$$

$$Peso = 7,97 \text{ kg}$$

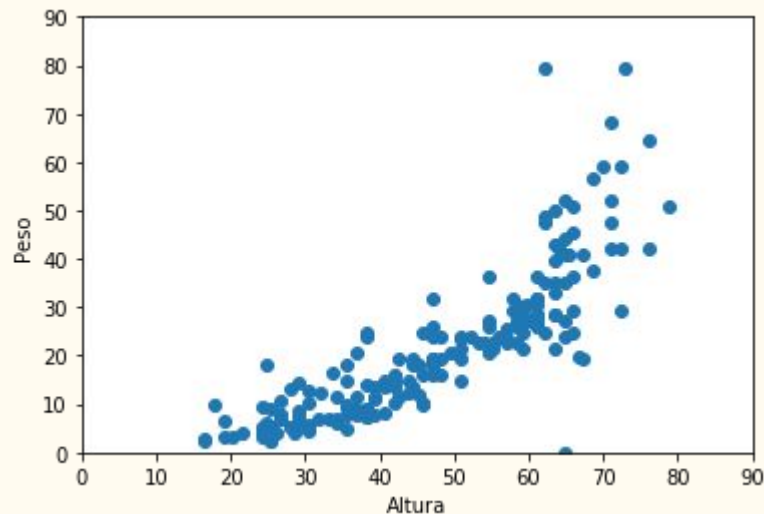
```
lr.predict(X)
```

# E se o problema não for linear?

É bem possível que os dados relacionados a uma tarefa não estejam distribuídos de forma que ocorra uma relação linear entre a variável alvo e a variável explicativa

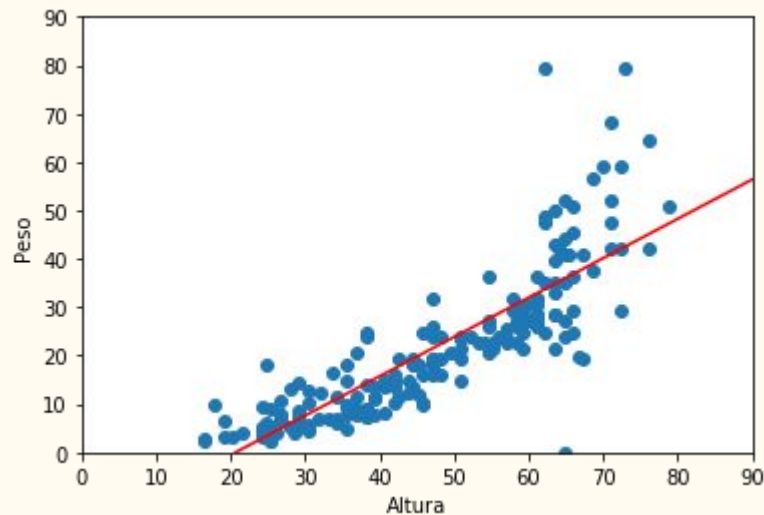
No exemplo ao lado, parece haver uma relação exponencial entre *Peso* e *Altura*

Fonte: <https://tmfilho.github.io/akcdata/>



# E se o problema não for linear?

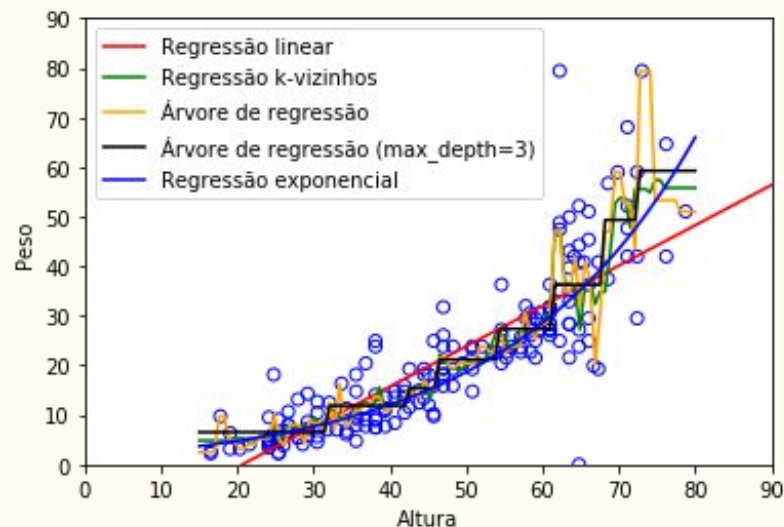
Como resultado, a regressão linear não apresenta o melhor ajuste



# E se o problema não for linear?

Existe uma grande variedade de métodos que podem ser usados nessa situação, incluindo árvores de decisão e k-vizinhos adaptados para regressão

Além disso, os dados parecem mostrar uma relação exponencial

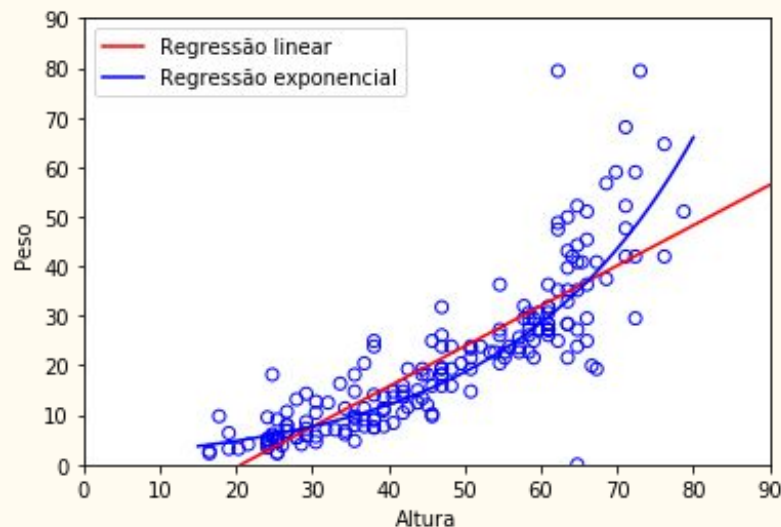


# E se o problema não for linear?

Existe uma grande variedade de métodos que podem ser usados nessa situação, incluindo árvores de decisão e k-vizinhos adaptados para regressão

Além disso, os dados parecem mostrar uma relação exponencial

Esse é um bom momento para aprender a diferença entre modelos paramétricos e não-paramétricos



Posso usar  
regressão em uma  
tarefa de  
classificação?

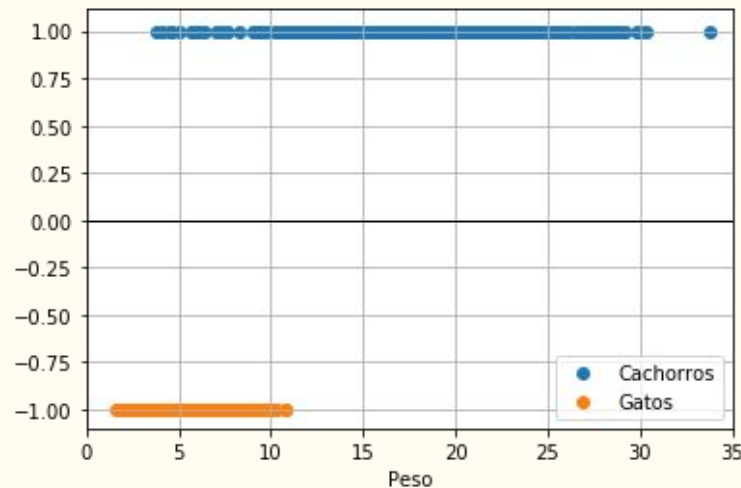


# Posso usar regressão em uma tarefa de classificação?

Claro, não tem ninguém segurando sua mão ;)

Para isso, se escolhe uma classe positiva e uma classe negativa

Aqui, Cachorro = 1 e Gato = -1



# Posso usar regressão em uma tarefa de classificação?

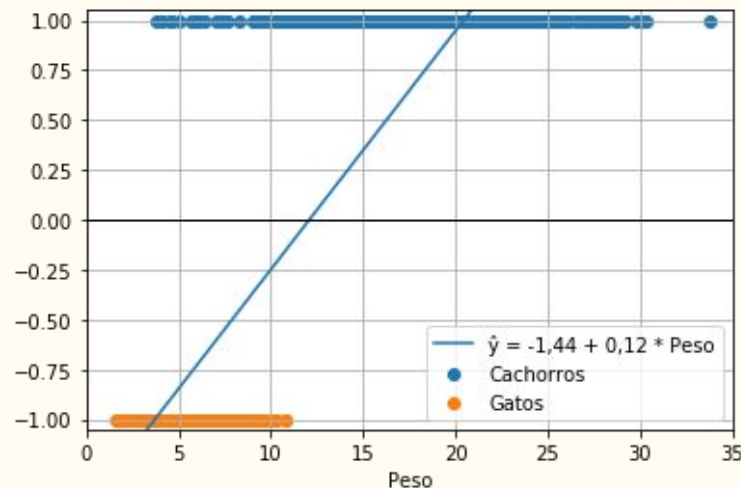
Com esses valores numéricos representando as classes, podemos ajustar um modelo de regressão linear

$$\hat{y} = -1,44 + 0,12 * \text{Peso}$$

Onde  $\hat{y}$  é um valor que indica a classe de um novo exemplo

$$\hat{y} > 0 \rightarrow \text{Cachorro}$$

$$\hat{y} < 0 \rightarrow \text{Gato}$$

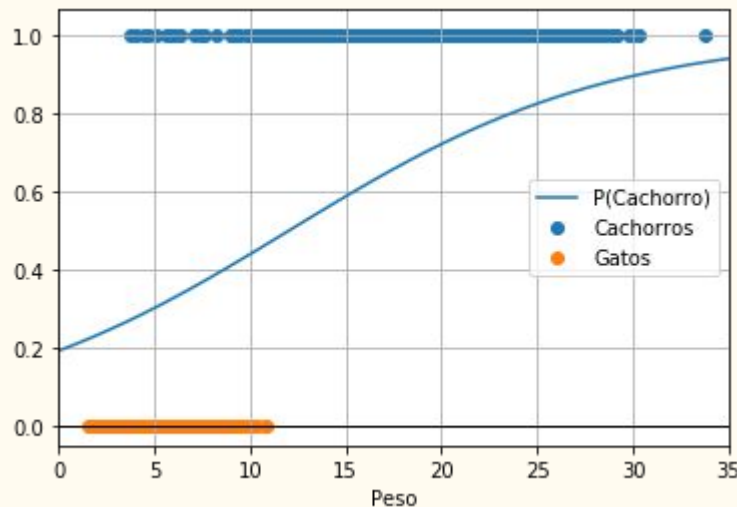


# E se eu quiser saber a probabilidade de um exemplo ser cachorro?

Podemos usar o modelo linear e aplicar função logística:

$$P(\text{Cachorro}) = \frac{1}{1 + e^{-(-1,44 + 0,12 * \text{Peso})}}$$

Este modelo é a famosa Regressão Logística e ele é a base teórica das redes neurais desenvolvidas para classificação



Como cada  
modelo é treinado  
em uma frase

# Support Vector Machine

Quais cachorros mais parecem com gatos e quais gatos mais parecem com cachorros, mas eu ainda tenho certeza do que eles são?

## Naïve Bayes

Qual a chance de um cachorro ou gato ter 47cm de altura e 100cm de comprimento e pesar 20kg?

# Rede Neural Artificial

Quais áreas do cérebro ficam mais ativas ao ver um cachorro ou um gato?

## Learning Vector Quantization

Gatos e cachorros elegeram representantes e eles decidem quem é cachorro ou gato.

# Random Forest

Meu conjunto é muito complexo para uma árvore, então vou dividir para conquistar e treinar um bocado delas.

## (Gradient ou Ada) Boosting

Minha primeira árvore erra mais nesses exemplos, então vou treinar uma árvore especializada neles (Repita).





# BATE-PAPO COM BRUNO BARUFALDI RADIOLOGIA E IA

01/07- 19:00

YOUTUBE

CANAL

ARIA - UFPB

GRADUADO EM CIÊNCIA DA  
COMPUTAÇÃO, MESTRE EM  
INFORMÁTICA NA UFPB E DOUTOR  
EM CIÊNCIAS NA USP. FEZ PÓS-  
DOUTORADO EM ENGENHARIA  
BIOMÉDICA E, ATUALMENTE, É  
PESQUISADOR NO DEPARTAMENTO  
DE RADIOLOGIA NA UNIVERSIDADE  
DA PENSILVÂNIA.

